

9.3

*IBM MQ Desenvolvendo Referência de  
Aplicativos*

**IBM**

**Nota**

Antes de usar estas informações e o produto que elas suportam, leia as informações em [“Avisos” na página 2273](#).

Esta edição se aplica à versão 9 liberação 3 do IBM® MQ e a todas as liberações e modificações subsequentes até que seja indicado de outra forma em novas edições.

Ao enviar informações para a IBM, você concede à IBM um direito não exclusivo de usar ou distribuir as informações da maneira que julgar apropriada, sem incorrer em qualquer obrigação para com você

© **Copyright International Business Machines Corporation 2007, 2024.**

# Índice

<b>Desenvolvendo a Referência do Aplicativo.....</b>	<b>7</b>
Referência de aplicativos MQI.....	7
Exemplos de código.....	8
Constantes.....	61
Tipos de dados usados no MQI.....	237
Chamadas de função.....	648
atributos de objetos.....	827
Códigos de retorno.....	905
Regras para validar opções de MQI.....	906
Mensagens de comando de publicação / assinatura enfileirados.....	909
Codificações da máquina.....	932
Opções de relatório e sinalizações de mensagem.....	935
saída de conversão de dados.....	939
Propriedades especificadas como elementos MQRFH2.....	963
Conversão de página de códigos.....	971
Padrões de codificação em plataformas de 64 bits.....	1025
IBM i Referência de programação de aplicativos (ILE/RPG).....	1029
Descrições de tipo de dados em IBM i.....	1030
Chamadas de função em IBM i.....	1295
Atributos de objetos no IBM i.....	1416
Aplicativos.....	1464
Códigos de retorno para IBM i (ILE RPG).....	1477
Regras para validar opções MQI para IBM i (ILE RPG).....	1479
Codificações da máquina em IBM i.....	1481
Opções de relatório e sinalizações de mensagem no IBM i.....	1484
Conversão de dados no IBM i.....	1488
Processamento de conversão em IBM i.....	1488
Convenções de processamento em IBM i.....	1490
Conversão de mensagens de relatório em IBM i.....	1494
MQDXP (parâmetro de saída de conversão de dados) no IBM i.....	1495
MQXCNCV (Converter caracteres) no IBM i.....	1500
MQCONVX (saída de conversão de Dados) em IBM i.....	1505
Saídas de usuário, saídas de API e referência de serviços instaláveis.....	1508
Estrutura MQIEP.....	1509
Referência de saída de conversão de dados.....	1512
MQ_PUBLISH_EXIT - saída Publish.....	1516
Chamadas de Saída do Canal e Estrutura de Dados.....	1524
Chamada de Saída de Carga de Trabalho do Cluster e Estruturas de Dados.....	1590
Referência de saída de API.....	1616
Informações de referência da interface de serviços instaláveis.....	1678
Informações de referência da interface de serviços instaláveis no IBM i.....	1742
As classes e interfaces do IBM MQ .NET.....	1783
Classe MQAsyncStatus.NET.....	1783
Classe MQAuthenticationInformationRecord.NET.....	1784
Classe MQDestination.NET.....	1785
Classe MQEnvironment.NET.....	1788
Classe MQException.NET.....	1790
Classe MQGetMessageOptions.NET.....	1791
Classe MQManagedObject.NET.....	1794
Classe MQMessage.NET.....	1796
Classe MQProcess.NET.....	1809
Classe MQPropertyDescriptor.NET.....	1811

Classe MQPutMessageOptions.NET.....	1813
Classe MQQueue.NET.....	1815
Classe MQQueueManager.NET.....	1823
Classe MQSubscription.NET.....	1836
Classe MQTopic.NET.....	1837
interface do IMQObjectTrigger.NET.....	1843
interface do MQC.NET.....	1844
Identificadores do conjunto de caracteres para aplicativos .NET.....	1844
Classes IBM MQ C++.....	1847
Referência cruzada de C++ e MQI.....	1848
Classe C++ do Registro ImqAuthentication.....	1865
Classe C++ ImqBinary.....	1867
Classe C++ ImqCache.....	1869
Classe C++ ImqChannel.....	1872
Classe C++ ImqCICSBridgeHeader.....	1878
ImqDeadLetterHeader Classe C++.....	1884
ImqDistributionListar classe C++.....	1887
Classe C++ ImqError.....	1888
Classe C++ ImqGetMessageOptions.....	1889
Classe C++ ImqHeader.....	1893
Classe C++ ImqIMSBridgeHeader.....	1894
Classe C++ ImqItem.....	1897
Classe C++ ImqMessage.....	1899
Classe C++ do Rastreador ImqMessage.....	1906
Classe C++ ImqNamelist.....	1909
Classe C++ ImqObject.....	1910
Classe C++ ImqProcess.....	1916
Classe C++ ImqPutMessageOptions.....	1918
Classe C++ ImqQueue.....	1920
Classe C++ do Gerenciador ImqQueue.....	1931
Classe C++ do cabeçalho ImqReference.....	1947
Classe C++ ImqString.....	1950
Classe C++ ImqTrigger.....	1956
Classe C++ do cabeçalho ImqWork.....	1958
Propriedades de objetos IBM MQ classes for JMS.....	1960
Dependências entre as propriedades de objetos do IBM MQ classes for JMS.....	1964
APPLICATIONNAME.....	1966
ASYNCEXCEPTION.....	1966
BALOPTIONS.....	1967
BALTYPE.....	1968
BALTIMEOUT.....	1969
BROKERCCDURSUBQ.....	1969
BROKERCCSUBQ.....	1970
BROKERCONQ.....	1970
BROKERDURSUBQ.....	1971
BROKERPUBQ.....	1971
BROKERPUBQMGR.....	1971
BROKERQMGR.....	1972
BROKERSUBQ.....	1972
BROKERVER.....	1973
CCDTURL.....	1973
CCSID.....	1974
CHANNEL.....	1974
CLEANUP.....	1975
CLEANUPINT.....	1975
connectionNameList.....	1976
CLIENTRECONNECTOPTIONS.....	1976
CLIENTRECONNECTTIMEOUT.....	1977





CLIENTID.....	1978
CLONESUPP.....	1978
COMPHDR.....	1979
COMPMSG.....	1979
CONNOPT.....	1980
CONNTAG.....	1981
DESCRIÇÃO.....	1981
DIRECTAUTH.....	1981
ENCODING.....	1982
EXPIRY.....	1983
FAILIFQUIESCE.....	1983
HOSTNAME.....	1984
LOCALADDRESS.....	1985
MAPNAMESTYLE.....	1985
MAXBUFFSIZE.....	1986
MDREAD.....	1986
MDWRITE.....	1987
MDMSGCTX.....	1987
MSGBATCHSZ.....	1988
MSGBODY.....	1989
MSGRETENTION.....	1989
MSGSELECTION.....	1990
MULTICAST.....	1990
OPTIMISTICPUBLICATION.....	1991
OUTCOMENOTIFICATION.....	1991
PERSISTENCE.....	1992
POLLINGINT.....	1993
PORT.....	1993
PRIORITY.....	1994
PROCESSDURATION.....	1994
PROVIDERVERSION.....	1995
PROXYHOSTNAME.....	1997
PROXYPORT.....	1997
PUBACKINT.....	1998
PUTASYNCALLOWED.....	1998
QMANAGER.....	1999
FILA.....	1999
READAHEADALLOWED.....	2000
READAHEADCLOSEPOLICY.....	2000
RECEIVECCSID.....	2001
RECEIVECONVERSION.....	2001
RECEIVEISOLATION.....	2002
RECEXIT.....	2002
RECEXITINIT.....	2003
REPLYTOSTYLE.....	2003
RESCANINT.....	2004
SECEXIT.....	2004
SECEXITINIT.....	2005
SENDCHECKCOUNT.....	2005
SENDEXIT.....	2006
SENDEXITINIT.....	2006
SHARECONVALLOWED.....	2007
SPARSESUBS.....	2008
SSLCIPHERSUITE.....	2008
SSLCRL.....	2009
SSLFIPSREQUIRED.....	2009
SSLPEERNAME.....	2010
SSLRESETCOUNT.....	2010

STATREFRESHINT.....	2010
SUBSTORE.....	2011
SYNCPOINTALLGETS.....	2011
TARGCLIENT.....	2012
TARGCLIENTMATCHING.....	2012
TEMPMODEL.....	2013
TEMPQPREFIX.....	2013
TEMPTOPICPREFIX.....	2014
tçpico.....	2014
TRANSPORT.....	2015
WILDCARDFORMAT.....	2015
A propriedade ENCODING.....	2016
Propriedades de TLS de objetos do JMS.....	2017
Referência do IBM MQ Message Service Client (XMS) for .NET.....	2018
.NET interfaces.....	2018
Propriedades de objetos XMS.....	2100
Managed File Transfer Desenvolvendo referência de aplicativos.....	2168
Exemplos de Uso de fteCreateTransfer para Iniciar Programas.....	2168
<b>fteAnt</b> : executar tarefas Ant no MFT.....	2170
MFT saídas de usuário para referência de customização.....	2196
Formatos de mensagens para mensagens que podem ser colocadas na fila de comandos do MFT Agent.....	2238
Referência do sistema de mensagens da REST API.....	2238
REST API recursos.....	2238
<b>Avisos.....</b>	<b>2273</b>
Informações sobre a Interface de Programação.....	2274
Marcas comerciais.....	2275

## Desenvolvendo a Referência do Aplicativo

---

Use os links fornecidos nesta seção para ajudar a desenvolver seus aplicativos IBM MQ .

- [“Referência de aplicativos MQI.” na página 7](#)
-  [“IBM i Referência de programação de aplicativos \(ILE/RPG\)” na página 1029](#)
-  [“Conversão de dados no IBM i” na página 1488](#)
- [“Saídas de usuário, saídas de API e referência de serviços instaláveis” na página 1508](#)
- [“As classes e interfaces do IBM MQ .NET” na página 1783](#)
- [“Classes IBM MQ C++” na página 1847](#)
- [“Propriedades de objetos IBM MQ classes for JMS” na página 1960](#)
- [“Referência do sistema de mensagens da REST API” na página 2238](#)

### Tarefas relacionadas

[Desenvolvendo Aplicativos](#)

### Referências relacionadas

[Classes do IBM MQ para bibliotecas Java](#)

[Classes do IBM MQ for JMS](#)

## Referência de aplicativos MQI.

---

Use os links fornecidos nesta seção para ajudar a desenvolver seus aplicativos Message Queue Interface (MQI).

- [“Exemplos de código” na página 8](#)
- [“Constantes” na página 61](#)
- [“Tipos de dados usados no MQI” na página 237](#)
- [“Chamadas de função” na página 648](#)
- [“atributos de objetos” na página 827](#)
- [“Códigos de retorno” na página 905](#)
- [“Regras para validar opções de MQI” na página 906](#)
- [“Codificações da máquina” na página 932](#)
- [“Opções de relatório e sinalizações de mensagem” na página 935](#)
- [“saída de conversão de dados” na página 939](#)
- [“Propriedades especificadas como elementos MQRFH2” na página 963](#)
- [“Conversão de página de códigos” na página 971](#)

### Conceitos relacionados

[“Saídas de usuário, saídas de API e referência de serviços instaláveis” na página 1508](#)

Use as informações nesta seção para ajudá-lo a desenvolver suas saídas de usuário, saídas de API e aplicativos de serviços instaláveis:

### Tarefas relacionadas

[Desenvolvendo Aplicativos](#)

### Referências relacionadas

[“As classes e interfaces do IBM MQ .NET” na página 1783](#)

As classes e interfaces do IBM MQ .NET são listadas alfabeticamente. As propriedades, métodos e construtores são descritos.

[“Classes IBM MQ C++” na página 1847](#)

As classes C++ IBM MQ encapsulam o Message Queue Interface (MQI) IBM MQ . Há um único arquivo de cabeçalho C++, **imqi.hpp**, que abrange todas essas classes.

[As bibliotecas IBM MQ Classes para Java](#)

[Classes de IBM MQ para JMS](#)

## Exemplos de código

Utilize as informações de referência desta seção para executar as tarefas que determinam as necessidades de negócios.

### Exemplos de linguagem C

Essa coleção de tópicos é principalmente obtida dos aplicativos de amostra do IBM MQ for z/OS . Eles são aplicáveis a todas as plataformas, exceto onde indicado.

#### ***Conectando-se a um gerenciador de filas***

Este exemplo demonstra como usar a chamadas MQCONN para conectar um programa a um gerenciador de filas em lote z/OS

Essa extração é obtida do aplicativo de amostra Browse (programa CSQ4BCA1) fornecido com IBM MQ for z/OS. Para obter os nomes e os locais dos aplicativos de amostra em outras plataformas, consulte [Programas processuais de amostra \(plataformas, exceto z/OS\)](#).

```
#include <cmqc.h>
:
static char Parm1[MQ_Q_MGR_NAME_LENGTH] ;

int main(int argc, char *argv[] )
{
    /*                                     */
    /* Variables for MQ calls             */
    /*                                     */
    MQHCONN Hconn;      /* Connection handle */
    MQLONG  CompCode;   /* Completion code  */
    MQLONG  Reason;     /* Qualifying reason */

    /* Copy the queue manager name, passed in the */
    /* parm field, to Parm1                        */
    strncpy(Parm1,argv[1],MQ_Q_MGR_NAME_LENGTH);

    /*                                     */
    /* Connect to the specified queue manager.    */
    /* Test the output of the connect call. If the */
    /* call fails, print an error message showing the */
    /* completion code and reason code, then leave the */
    /* program.                                     */
    /*                                     */
    MQCONN(Parm1,
           &Hconn,
           &CompCode,
           &Reason);
    if ((CompCode != MQCC_OK) | (Reason != MQRC_NONE))
    {
        sprintf(pBuff, MESSAGE_4_E,
                ERROR_IN_MQCONN, CompCode, Reason);
        PrintLine(pBuff);
        RetCode = CSQ4_ERROR;
        goto AbnormalExit2;
    }
    :
}
```

#### ***desconectando-se de um gerenciador de filas***

Este exemplo demonstra como usar a chamada MQDISC para desconectar um programa de um gerenciador de fila em lote do z/OS

As variáveis usadas nessa extração de códigos são aquelas que foram configuradas no “Conectando-se a um gerenciador de filas” na página 8 Essa extração é obtida do aplicativo de amostra Browse (programa CSQ4BCA1) fornecido com IBM MQ for z/OS. Para obter os nomes e os locais dos aplicativos de amostra em outras plataformas, consulte [Programas processuais de amostra \(plataformas, exceto z/OS\)](#).

```

:
/*                                     */
/* Disconnect from the queue manager. Test the */
/* output of the disconnect call. If the call */
/* fails, print an error message showing the */
/* completion code and reason code.         */
/*                                     */
MQDISC(&Hconn,
      &CompCode,
      &Reason);
if ((CompCode != MQCC_OK) || (Reason != MQRC_NONE))
{
    sprintf(pBuff, MESSAGE_4_E,
           ERROR_IN_MQDISC, CompCode, Reason);
    PrintLine(pBuff);
    RetCode = CSQ4_ERROR;
}
:

```

### ***Criando uma fila dinâmica***

Este exemplo demonstra como usar a chamada MQOPEN para criar uma fila dinâmica..

Essa extração é obtida do aplicativo de amostra Mail Manager (programa CSQ4TCD1) fornecido com IBM MQ for z/OS. Para obter os nomes e os locais dos aplicativos de amostra em outras plataformas, consulte [Programas processuais de amostra \(plataformas, exceto z/OS\)](#).

```

:
MQLONG HCONN = 0; /* Connection handle */
MQHOBJ HOBJ; /* MailQ Object handle */
MQHOBJ HobjTempQ; /* TempQ Object Handle */
MQLONG CompCode; /* Completion code */
MQLONG Reason; /* Qualifying reason */
MQOD ObjDesc = {MQOD_DEFAULT};
MQLONG OpenOptions; /* Options control MQOPEN */

/*-----*/
/* Initialize the Object Descriptor (MQOD) */
/* control block. (The remaining fields */
/* are already initialized.) */
/*-----*/
strncpy( ObjDesc.ObjectName,
        SYSTEM_REPLY_MODEL,
        MQ_Q_NAME_LENGTH );
strncpy( ObjDesc.DynamicQName,
        SYSTEM_REPLY_INITIAL,
        MQ_Q_NAME_LENGTH );
OpenOptions = MQOO_INPUT_AS_Q_DEF;
/*-----*/
/* Open the model queue and, therefore, */
/* create and open a temporary dynamic */
/* queue */
/*-----*/
MQOPEN( HCONN,
        &ObjDesc,
        OpenOptions,
        &HobjTempQ,
        &CompCode,
        &Reason );
if ( CompCode == MQCC_OK ) {
}
else {
    /*-----*/
    /* Build an error message to report the */
    /* failure of the opening of the model */
    /* queue */
    /*-----*/
    MQMErrorHandling( "OPEN TEMPQ", CompCode,

```

```

        Reason );
    ErrorFound = TRUE;
}
return ErrorFound;
}

```

## Abrindo uma fila existente

Este exemplo demonstra como usar a chamada MQOPEN para abrir uma fila já definida.

Essa extração é obtida do aplicativo de amostra Browse (programa CSQ4BCA1) fornecido com IBM MQ for z/OS. Para obter os nomes e os locais dos aplicativos de amostra em outras plataformas, consulte [Programas processuais de amostra \(plataformas, exceto z/OS\)](#).

```

#include <cmqc.h>
...
static char Parm1[MQ_Q_MGR_NAME_LENGTH];
...
int main(int argc, char *argv[] )
{
    /*
    /*     Variables for MQ calls                               */
    /*
    MQHCONN Hconn ;           /* Connection handle           */
    MQLONG  CompCode;         /* Completion code     */
    MQLONG  Reason;          /* Qualifying reason   */
    MQOD    ObjDesc = { MQOD_DEFAULT };
    MQLONG  OpenOptions;     /* Options that control */
    /* the MQOPEN call   */
    MQHOBJ  Hobj;           /* Object handle       */
    ...
    /* Copy the queue name, passed in the parm field,       */
    /* to Parm2 strncpy(Parm2,argv[2],                      */
    /* MQ_Q_NAME_LENGTH);                                  */
    ...
    /*
    /* Initialize the object descriptor (MQOD) control     */
    /* block. (The initialization default sets StrucId,    */
    /* Version, ObjectType, ObjectQMgrName,              */
    /* DynamicQName, and AlternateUserid fields)         */
    /*
    strncpy(ObjDesc.ObjectName,Parm2,MQ_Q_NAME_LENGTH);
    ...
    /* Initialize the other fields required for the open   */
    /* call (Hobj is set by the MQCONN call).             */
    /*
    OpenOptions = MQOO_BROWSE;
    ...
    /*
    /* Open the queue.                                     */
    /* Test the output of the open call. If the call      */
    /* fails, print an error message showing the         */
    /* completion code and reason code, then bypass     */
    /* processing, disconnect and leave the program.     */
    /*
    MQOPEN(Hconn,
           &ObjDesc,
           OpenOptions,
           &Hobj,
           &CompCode,
           &Reason);

    if ((CompCode != MQCC_OK) || (Reason != MQRC_NONE))
    {
        sprintf(pBuff, MESSAGE_4_E,
                ERROR_IN_MQOPEN, CompCode, Reason);
        PrintLine(pBuff);
        RetCode = CSQ4_ERROR;
        goto AbnormalExit1; /* disconnect processing */
    }
    ...
} /* end of main */

```

## fechar uma fila

Este exemplo demonstra como usar a chamada MQCLOSE para fechar uma fila


Essa extração é obtida do aplicativo de amostra Browse (programa CSQ4BCA1) fornecido com IBM MQ for z/OS. Para obter os nomes e os locais dos aplicativos de amostra em outras plataformas, consulte [Programas processuais de amostra \(plataformas, exceto z/OS\)](#).

```
:
/*                                     */
/* Close the queue.                    */
/* Test the output of the close call.  If the call */
/* fails, print an error message showing the */
/* completion code and reason code.      */
/*                                     */
MQCLOSE(Hconn,
        &Hobj,
        MQCO_NONE,
        &CompCode,
        &Reason);
if ((CompCode != MQCC_OK) || (Reason != MQRC_NONE))
{
    sprintf(pBuff, MESSAGE_4_E,
            ERROR_IN_MQCLOSE, CompCode, Reason);
    PrintLine(pBuff);
    RetCode = CSQ4_ERROR;
}
:
```

## Colocando uma mensagem usando MQPUT

Este exemplo demonstra como usar a chamada MQPUT para colocar uma mensagem em uma fila

Essa extração não é obtida dos aplicativos de amostra fornecidos com o IBM MQ. Para os nomes e locais dos aplicativos de amostra, consulte [Programas processuais de amostra \(plataformas, exceto z/OS\)](#)

 e [Programas de amostra para IBM MQ for z/OS](#).

```
:
qput()
{
    MQMD    MsgDesc;
    MQPMO   PutMsgOpts;
    MQLONG  CompCode;
    MQLONG  Reason;
    MQHCONN Hconn;
    MQHOBJ  Hobj;
    char message_buffer[] = "MY MESSAGE";
    /*-----*/
    /* Set up PMO structure.                    */
    /*-----*/
    memset(&PutMsgOpts, '\0', sizeof(PutMsgOpts));
    memcpy(PutMsgOpts.StrucId, MQPMO_STRUC_ID,
           sizeof(PutMsgOpts.StrucId));
    PutMsgOpts.Version = MQPMO_VERSION_1;
    PutMsgOpts.Options = MQPMO_SYNCPOINT;

    /*-----*/
    /* Set up MD structure.                    */
    /*-----*/
    memset(&MsgDesc, '\0', sizeof(MsgDesc));
    memcpy(MsgDesc.StrucId, MQMD_STRUC_ID,
           sizeof(MsgDesc.StrucId));
    MsgDesc.Version      = MQMD_VERSION_1;
    MsgDesc.Expiry       = MQEI_UNLIMITED;
    MsgDesc.Report       = MQRO_NONE;
    MsgDesc.MsgType      = MQMT_DATAGRAM;
    MsgDesc.Priority     = 1;
    MsgDesc.Persistence  = MQPER_PERSISTENT;
    memset(MsgDesc.ReplyToQ,
           '\0',
           sizeof(MsgDesc.ReplyToQ));
    /*-----*/
    /* Put the message.                        */
    /*-----*/
    MQPUT(Hconn, Hobj, &MsgDesc, &PutMsgOpts,
```

```
sizeof(message_buffer), message_buffer,
&CompCode, &Reason);
```

```

/*-----*/
/* Check completion and reason codes. */
/*-----*/
switch (CompCode)
{
  case MQCC_OK:
    break;
  case MQCC_FAILED:
    switch (Reason)
    {
      case MQRC_Q_FULL:
      case MQRC_MSG_TOO_BIG_FOR_Q:
        break;
      default:
        break; /* Perform error processing */
    }
    break;
  default:
    break; /* Perform error processing */
}
}
}

```

### Colocando uma mensagem usando MQPUT1

Este exemplo demonstra como usar a chamada MQPUT1 para abrir uma fila, colocar uma única mensagem na fila, em seguida, fechar a fila.

Essa extração é obtida do aplicativo de amostra Credit Check (programa CSQ4CCB5) fornecido com IBM MQ for z/OS. Para obter os nomes e os locais dos aplicativos de amostra em outras plataformas, consulte [Programas processuais de amostra \(plataformas, exceto z/OS\)](#).

```

:
MQLONG Hconn; /* Connection handle */
MQHOBJ Hobj_CheckQ; /* Object handle */
MQLONG CompCode; /* Completion code */
MQLONG Reason; /* Qualifying reason */
MQOD ObjDesc = {MQOD_DEFAULT}; /* Object descriptor */
MQMD MsgDesc = {MQMD_DEFAULT}; /* Message descriptor */
MQLONG OpenOptions; /* Control the MQOPEN call */

MQGMO GetMsgOpts = {MQGMO_DEFAULT}; /* Get Message Options */
MQLONG MsgBufLen; /* Length of message buffer */
CSQ4BCAQ MsgBuffer; /* Message structure */
MQLONG DataLen; /* Length of message */

MQPMO PutMsgOpts = {MQPMO_DEFAULT}; /* Put Message Options */
CSQ4BQRM PutBuffer; /* Message structure */
MQLONG PutBufLen = sizeof(PutBuffer); /* Length of message buffer */
:

```

```

void Process_Query(void)
{
  /*
  /* Build the reply message
  /*
  /*
  /* Set the object descriptor, message descriptor and
  /* put message options to the values required to
  /* create the reply message.
  /*
  /*
  strncpy(ObjDesc.ObjectName, MsgDesc.ReplyToQ,
          MQ_Q_NAME_LENGTH);
  strncpy(ObjDesc.ObjectQMGrName, MsgDesc.ReplyToQMGr,
          MQ_Q_MGR_NAME_LENGTH);
  MsgDesc.MsgType = MQMT_REPLY;

```



```

MsgDesc.Report = MQRO_NONE;
memset(MsgDesc.ReplyToQ, ' ', MQ_Q_NAME_LENGTH);
memset(MsgDesc.ReplyToQMgr, ' ', MQ_Q_MGR_NAME_LENGTH);
memcpy(MsgDesc.MsgId, MQMI_NONE, sizeof(MsgDesc.MsgId));
PutMsgOpts.Options = MQPMO_SYNCPOINT +
                    MQPMO_PASS_IDENTITY_CONTEXT;
PutMsgOpts.Context = Hobj_CheckQ;
PutBufLen = sizeof(PutBuffer);
MQPUT1(Hconn,
        &ObjDesc,
        &MsgDesc,
        &PutMsgOpts,
        PutBufLen,
        &PutBuffer,
        &CompCode,
        &Reason);

if (CompCode != MQCC_OK)
{
    strncpy(TS_Operation, "MQPUT1",
            sizeof(TS_Operation));
    strncpy(TS_ObjName, ObjDesc.ObjectName,
            MQ_Q_NAME_LENGTH);
    Record_Call_Error();
    Forward_Msg_To_DLQ();
}
return;
}
:

```

### Obtendo uma mensagem

Este exemplo demonstra como usar a chamada MQGET para remover uma mensagem de uma fila.

Essa extração é obtida do aplicativo de amostra Browse (programa CSQ4BCA1) fornecido com IBM MQ for z/OS. Para obter os nomes e os locais dos aplicativos de amostra em outras plataformas, consulte [Programas processuais de amostra \(plataformas, exceto z/OS\)](#).

```

#include "cmqc.h"
:
#define BUFFERLENGTH 80
:
int main(int argc, char *argv[] )
{
    /*                                     */
    /*   Variables for MQ calls           */
    /*                                     */
    MQHCONN Hconn ;                       /* Connection handle   */
    MQLONG  CompCode;                      /* Completion code     */
    MQLONG  Reason;                        /* Qualifying reason   */
    MQHOBJ  Hobj;                          /* Object handle       */
    MQMD    MsgDesc = { MQMD_DEFAULT };    /* Message descriptor  */
    MQLONG  DataLength ;                   /* Length of the message */
    MQCHAR  Buffer[BUFFERLENGTH+1];        /* Area for message data */
    MQGMO   GetMsgOpts = { MQGMO_DEFAULT }; /* Options which control */
    /*                                     */
    /*   the MQGET call                   */
    MQLONG  BufferLength = BUFFERLENGTH ;   /* Length of buffer    */
    :
    /*   No need to change the message descriptor */
    /*   (MQMD) control block because initialization */
    /*   default sets all the fields.             */
    /*                                     */
    /*   Initialize the get message options (MQGMO) */
    /*   control block (the copy file initializes all */
    /*   the other fields).                       */
    /*                                     */
    GetMsgOpts.Options = MQGMO_NO_WAIT      +
                        MQGMO_BROWSE_FIRST +
                        MQGMO_ACCEPT_TRUNCATED_MSG;

    /*                                     */
    /* Get the first message.               */
    /* Test for the output of the call is carried out */
    /* in the 'for' loop.                   */
    /*                                     */
}

```

```

MQGET(Hconn,
      Hobj,
      &MsgDesc,
      &GetMsgOpts,
      BufferLength,
      Buffer,
      &DataLength,
      &CompCode,
      &Reason);

```

```

/*                                     */
/* Process the message and get the next message, */
/* until no messages remaining.                */
/*                                     */
/* If the call fails for any other reason,      */
/* print an error message showing the completion */
/* code and reason code.                      */
/*                                     */
if ( (CompCode == MQCC_FAILED) &&
     (Reason == MQRC_NO_MSG_AVAILABLE) )
{
    ...
}
else
{
    sprintf(pBuff, MESSAGE_4_E,
           ERROR_IN_MQGET, CompCode, Reason);
    PrintLine(pBuff);
    RetCode = CSQ4_ERROR;
    ...
}
} /* end of main */

```

### ***Obtendo uma mensagem usando a opção de espera***

Este exemplo demonstra como usar a opção de espera da chamada MQGET

Este código aceita mensagens truncadas.. Essa extração é obtida do aplicativo de amostra Credit Check (programa CSQ4CCB5) fornecido com IBM MQ for z/OS. Para obter os nomes e os locais dos aplicativos de amostra em outras plataformas, consulte [Programas processuais de amostra \(plataformas, exceto z/OS\)](#).

```

:
MQLONG  Hconn;           /* Connection handle */
MQHOBJ  Hobj_CheckQ;    /* Object handle     */
MQLONG  CompCode;       /* Completion code   */
MQLONG  Reason;        /* Qualifying reason */
MQOD    ObjDesc        = {MQOD_DEFAULT};
                          /* Object descriptor */
MQMD    MsgDesc        = {MQMD_DEFAULT};
                          /* Message descriptor */
MQLONG  OpenOptions;   /* Control the MQOPEN call */
MQGMO   GetMsgOpts     = {MQGMO_DEFAULT};
                          /* Get Message Options */
MQLONG  MsgBuffLen;    /* Length of message buffer */
CSQ4BCAQ MsgBuffer;    /* Message structure */
MQLONG  DataLen;       /* Length of message */

```

```

:
void main(void)
{
    ...
    /* Initialize options and open the queue for input */
    /*                                     */
    ...
    /* Get and process messages */
    /*                                     */
    GetMsgOpts.Options = MQGMO_WAIT +
                        MQGMO_ACCEPT_TRUNCATED_MSG +
                        MQGMO_SYNCPOINT;
    GetMsgOpts.WaitInterval = WAIT_INTERVAL;
}

```

```

MsgBufLen = sizeof(MsgBuffer);
memcpy(MsgDesc.MsgId, MQMI_NONE,
        sizeof(MsgDesc.MsgId));
memcpy(MsgDesc.CorrelId, MQCI_NONE,
        sizeof(MsgDesc.CorrelId));

/*
/* Make the first MQGET call outside the loop
/*
MQGET(Hconn,
      Hobj_CheckQ,
      &MsgDesc,
      &GetMsgOpts,
      MsgBufLen,
      &MsgBuffer,
      &DataLen,
      &CompCode,
      &Reason);
:
/*
/* Test the output of the MQGET call. If the call
/* failed, send an error message showing the
/* completion code and reason code, unless the
/* reason code is NO_MSG_AVAILABLE.
/*
if (Reason != MQRC_NO_MSG_AVAILABLE)
{
  strncpy(TS_Operation, "MQGET", sizeof(TS_Operation));
  strncpy(TS_ObjName, ObjDesc.ObjectName,
          MQ_Q_NAME_LENGTH);
  Record_Call_Error();
}
:

```

## Obtendo uma mensagem usando sinalização

A sinalização está disponível apenas com o IBM MQ for z/OS

Este exemplo demonstra como usar a chamada MQGET para configurar um sinal para que você seja notificado quando uma mensagem adequada chegar em uma fila. Essa extração não é obtida dos aplicativos de amostra fornecidos com o IBM MQ.

```

:
get_set_signal()
{
  MQMD    MsgDesc;
  MQGMO   GetMsgOpts;
  MQLONG  CompCode;
  MQLONG  Reason;
  MQHCONN Hconn;
  MQHOBJ  Hobj;
  MQLONG  BufferLength;
  MQLONG  DataLength;
  char message_buffer[100];
  long int q_ecn, work_ecn;
  short int signal_sw, endloop;
  long int mask = 255;

  /*-----*/
  /* Set up GMO structure.
  /*-----*/
  memset(&GetMsgOpts, '\0', sizeof(GetMsgOpts));
  memcpy(GetMsgOpts.StrucId, MQGMO_STRUC_ID,
         sizeof(GetMsgOpts.StrucId));
  GetMsgOpts.Version      = MQGMO_VERSION_1;
  GetMsgOpts.WaitInterval = 1000;
  GetMsgOpts.Options      = MQGMO_SET_SIGNAL +
                           MQGMO_BROWSE_FIRST;

  q_ecn = 0;
  GetMsgOpts.Signal1 = &q_ecn;
  /*-----*/
  /* Set up MD structure.
  /*-----*/
  memset(&MsgDesc, '\0', sizeof(MsgDesc));
  memcpy(MsgDesc.StrucId, MQMD_STRUC_ID,
         sizeof(MsgDesc.StrucId));
  MsgDesc.Version = MQMD_VERSION_1;
  MsgDesc.Report  = MQRO_NONE;
  memcpy(MsgDesc.MsgId, MQMI_NONE,

```

```

        sizeof(MsgDesc.MsgId));
memcpy(MsgDesc.CorrelId, MQCI_NONE,
        sizeof(MsgDesc.CorrelId));

```

```

/*-----*/
/* Issue the MQGET call. */
/*-----*/
BufferLength = sizeof(message_buffer);
signal_sw = 0;

MQGET(Hconn, Hobj, &MsgDesc, &GetMsgOpts,
      BufferLength, message_buffer, &DataLength,
      &CompCode, &Reason);
/*-----*/
/* Check completion and reason codes. */
/*-----*/
switch (CompCode)
{
    case (MQCC_OK):          /* Message retrieved */
        break;
    case (MQCC_WARNING):
        switch (Reason)
        {
            case (MQRC_SIGNAL_REQUEST_ACCEPTED):
                signal_sw = 1;
                break;
            default:
                break; /* Perform error processing */
        }
        break;
    case (MQCC_FAILED):
        switch (Reason)
        {
            case (MQRC_Q_MGR_NOT_AVAILABLE):
            case (MQRC_CONNECTION_BROKEN):
            case (MQRC_Q_MGR_STOPPING):
                break;
            default:
                break; /* Perform error processing. */
        }
        break;
    default:
        break; /* Perform error processing. */
}
/*-----*/
/* If the SET_SIGNAL was accepted, set up a loop to */
/* check whether a message has arrived at one second */
/* intervals. The loop ends if a message arrives or */
/* the wait interval specified in the MQGMO */
/* structure has expired. */
/* If a message arrives on the queue, another MQGET */
/* must be issued to retrieve the message. If other */
/* MQM calls have been made in the intervening */
/* period, this may necessitate reinitializing the */
/* MQMD and MQGMO structures. */
/* In this code, no intervening calls */
/* have been made, so the only change required to */
/* the structures is to specify MQGMO_NO_WAIT, */
/* since we now know the message is there. */
/* This code uses the EXEC CICS DELAY command to */
/* suspend the program for a second. A batch program */
/* may achieve the same effect by calling an */
/* assembler language subroutine which issues a */
/* z/OS STIMER macro. */
/*-----*/

```

```

if (signal_sw == 1)
{
    endloop = 0;
    do
    {
        EXEC CICS DELAY FOR HOURS(0) MINUTES(0) SECONDS(1);
        work_ecb = q_ecb & mask;
        switch (work_ecb)
        {
            case (MQEC_MSG_ARRIVED):

```

```

        endloop = 1;
        mqgmo_options = MQGMO_NO_WAIT;
        MQGET(Hconn, Hobj, &MsgDesc, &GetMsgOpts,
            BufferLength, message_buffer,
            &DataLength, &CompCode, &Reason);
        if (CompCode != MQCC_OK)
            ; /* Perform error processing. */
        break;
        case (MQEC_WAIT_INTERVAL_EXPIRED):
        case (MQEC_WAIT_CANCELED):
            endloop = 1;
            break;
        default:
            break;
    }
} while (endloop == 0);
}
return;
}
}

```

### **Consultando sobre os atributos de um objeto**

Este exemplo demonstra como usar a chamada MQINQ para consultar sobre os atributos de uma fila

Essa extração é obtida do aplicativo de amostra Atributos da Fila (programa CSQ4CCC1) fornecido com IBM MQ for z/OS. Para obter os nomes e os locais dos aplicativos de amostra em outras plataformas, consulte [Programas processuais de amostra \(plataformas, exceto z/OS\)](#).

```

#include <mqc.h> /* MQ API header file */
:
#define NUMBEROFSELECTORS 2

const MQHCONN Hconn = MQHC_DEF_HCONN;
:
static void InquireGetAndPut(char *Message,
    PMQHOBJ pHobj,
    char *Object)

{
/* Declare local variables */
/*
MQLONG SelectorCount = NUMBEROFSELECTORS;
/* Number of selectors */
MQLONG IntAttrCount = NUMBEROFSELECTORS;
/* Number of int attrs */
MQLONG CharAttrLength = 0;
/* Length of char attribute buffer */
MQCHAR *CharAttrs ;
/* Character attribute buffer */
MQLONG SelectorsTable[NUMBEROFSELECTORS];
/* attribute selectors */
MQLONG IntAttrsTable[NUMBEROFSELECTORS];
/* integer attributes */
MQLONG CompCode; /* Completion code */
MQLONG Reason; /* Qualifying reason */
/*
/* Open the queue. If successful, do the inquire */
/* call. */
/*
/*
/* Initialize the variables for the inquire */
/* call: */
/* - Set SelectorsTable to the attributes whose */
/* status is */
/* required */
/* - All other variables are already set */
/*
SelectorsTable[0] = MQIA_INHIBIT_GET;
SelectorsTable[1] = MQIA_INHIBIT_PUT;
/*
/* Issue the inquire call */
/* Test the output of the inquire call. If the */
/* call failed, display an error message */
/* showing the completion code and reason code, */
/* otherwise display the status of the */
/* INHIBIT-GET and INHIBIT-PUT attributes */
}

```

```

/*                                          */
MQINQ(Hconn,
      *pHobj,
      SelectorCount,
      SelectorsTable,
      IntAttrCount,
      IntAttrsTable,
      CharAttrLength,
      CharAttrs,
      &CompCode,
      &Reason);
if (CompCode != MQCC_OK)
{
    sprintf(Message, MESSAGE_4_E,
            ERROR_IN_MQINQ, CompCode, Reason);
    SetMsg(Message);
}
else
{
    /* Process the changes */
} /* end if CompCode */

```

### **Configurando os atributos de uma fila**

Este exemplo demonstra como usar a chamada MQSET para alterar os atributos de uma fila

Essa extração é obtida do aplicativo de amostra Atributos da Fila (programa CSQ4CCC1) fornecido com IBM MQ for z/OS. Para obter os nomes e os locais dos aplicativos de amostra em outras plataformas, consulte [Programas processuais de amostra \(plataformas, exceto z/OS\)](#).

```

#include <cmqc.h>      /* MQ API header file      */
:
#define NUMBEROFSELECTORS 2

const MQHCONN Hconn = MQHC_DEF_HCONN;

static void InhibitGetAndPut(char *Message,
                             PMQHOBJ pHobj,
                             char *Object)
{
    /*                                          */
    /*      Declare local variables          */
    /*                                          */
    /*                                          */
    MQLONG SelectorCount = NUMBEROFSELECTORS;
    /*                                          */
    /*      /* Number of selectors          */
    MQLONG IntAttrCount = NUMBEROFSELECTORS;
    /*                                          */
    /*      /* Number of int attrs          */
    MQLONG CharAttrLength = 0;
    /*                                          */
    /*      /* Length of char attribute buffer */
    MQCHAR *CharAttrs ;
    /*                                          */
    /*      /* Character attribute buffer          */
    MQLONG SelectorsTable[NUMBEROFSELECTORS];
    /*                                          */
    /*      /* attribute selectors          */
    MQLONG IntAttrsTable[NUMBEROFSELECTORS];
    /*                                          */
    /*      /* integer attributes          */
    MQLONG CompCode;
    /*                                          */
    /*      /* Completion code          */
    MQLONG Reason;
    /*                                          */
    /*      /* Qualifying reason          */
    :
    /*                                          */
    /*      /* Open the queue. If successful, do the */
    /*      /* inquire call.          */
    /*                                          */
    :
    /*                                          */
    /*      /* Initialize the variables for the set call: */
    /*      /* - Set SelectorsTable to the attributes to be */
    /*      /* set          */
    /*      /* - Set IntAttrsTable to the required status */
    /*      /* - All other variables are already set          */
    /*      /*                                          */
    /*      /*                                          */
    SelectorsTable[0] = MQIA_INHIBIT_GET;
    SelectorsTable[1] = MQIA_INHIBIT_PUT;
    IntAttrsTable[0] = MQQA_GET_INHIBITED;
    IntAttrsTable[1] = MQQA_PUT_INHIBITED;
    :
}

```

```

/*                                     */
/* Issue the set call.                 */
/* Test the output of the set call. If the */
/* call fails, display an error message */
/* showing the completion code and reason */
/* code; otherwise move INHIBITED to the */
/* relevant screen map fields         */
/*                                     */
MQSET(Hconn,
      *pHobj,
      SelectorCount,
      SelectorsTable,
      IntAttrCount,
      IntAttrsTable,
      CharAttrLength,
      CharAttrs,
      &CompCode,
      &Reason);
if (CompCode != MQCC_OK)
{
    sprintf(Message, MESSAGE_4_E,
            ERROR_IN_MQSET, CompCode, Reason);
    SetMsg(Message);
}
else
{
    /* Process the changes */
} /* end if CompCode */

```

## Recuperando informações de status com MQSTAT

Este exemplo demonstra como emitir um MQPUT assíncrono e recuperar as informações de status com MQSTAT

Essa extração é obtida do aplicativo de amostra MQSTAT de Chamada (programa amqsapt0) fornecido com os sistemas IBM MQ for Windows Para obter os nomes e os locais dos aplicativos de amostra em outras plataformas, consulte [Programas processuais de amostra \(plataformas, exceto z/OS\)](#).

```

/*****
/*                                     */
/* Program name: AMQSAPT0             */
/*                                     */
/* Description: Sample C program that asynchronously puts messages */
/* to a message queue (example using MQPUT & MQSTAT). */
/*                                     */
/* Licensed Materials - Property of IBM */
/*                                     */
/* 63H9336                             */
/* (c) Copyright IBM Corp. 2006, 2024. All Rights Reserved. */
/*                                     */
/* US Government Users Restricted Rights - Use, duplication or */
/* disclosure restricted by GSA ADP Schedule Contract with */
/* IBM Corp.                             */
/*                                     */
/*****
/* Function:                           */
/*                                     */
/* AMQSAPT0 is a sample C program to put messages on a message */
/* queue with asynchronous response option, querying the success */
/* of the put operations with MQSTAT. */
/*                                     */
/* -- messages are sent to the queue named by the parameter */
/*                                     */
/* -- gets lines from StdIn, and adds each to target */
/* queue, taking each line of text as the content */
/* of a datagram message; the sample stops when a null */
/* line (or EOF) is read. */
/* New-line characters are removed. */
/* If a line is longer than 99 characters it is broken up */
/* into 99-character pieces. Each piece becomes the */
/* content of a datagram message. */
/* If the length of a line is a multiple of 99 plus 1, for */
/* example, 199, the last piece will only contain a */
/* new-line character so will terminate the input. */
/*                                     */
/*****

```

```

/*      -- writes a message for each MQI reason other than          */
/*      MQRC_NONE; stops if there is a MQI completion code        */
/*      of MQCC_FAILED                                           */
/*                                                                */
/*      -- summarizes the overall success of the put operations   */
/*      through a call to MQSTAT to query MQSTAT_TYPE_ASYNC_ERROR*/
/*                                                                */
/*      Program logic:                                           */
/*      MQOPEN target queue for OUTPUT                            */
/*      while end of input file not reached,                     */
/*      . read next line of text                                  */
/*      . MQPUT datagram message with text line as data          */
/*      MQCLOSE target queue                                     */
/*      MQSTAT connection                                        */
/*                                                                */
/*                                                                */
/*****
/*
/*      AMQSAPT0 has the following parameters                      */
/*      required:                                                */
/*          (1) The name of the target queue                      */
/*      optional:                                                */
/*          (2) Queue manager name                               */
/*          (3) The open options                                  */
/*          (4) The close options                                */
/*          (5) The name of the target queue manager             */
/*          (6) The name of the dynamic queue                    */
/*                                                                */
/*****
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
    /* includes for MQI */
#include <cmqc.h>

int main(int argc, char **argv)
{
    /* Declare file and character for sample input                */
    FILE *fp;

    /* Declare MQI structures needed                              */
    MQOD    od = {MQOD_DEFAULT}; /* Object Descriptor          */
    MQMD    md = {MQMD_DEFAULT}; /* Message Descriptor     */
    MQPMO    pmo = {MQPMO_DEFAULT}; /* put message options    */
    MQSTS    sts = {MQSTS_DEFAULT}; /* status information     */
    /* note, sample uses defaults where it can */
    MQHCONN  Hcon; /* connection handle     */
    MQHOBJ   Hobj; /* object handle         */
    MQLONG   O_options; /* MQOPEN options       */
    MQLONG   C_options; /* MQCLOSE options      */
    MQLONG   CompCode; /* completion code       */
    MQLONG   OpenCode; /* MQOPEN completion code */
    MQLONG   Reason; /* reason code           */
    MQLONG   CReason; /* reason code for MQCONN */
    MQLONG   messlen; /* message length        */
    char     buffer[100]; /* message buffer        */
    char     QMName[50]; /* queue manager name    */

    printf("Sample AMQSAPT0 start\n");
    if (argc < 2)
    {
        printf("Required parameter missing - queue name\n");
        exit(99);
    }

    /*****
    /*
    /*      Connect to queue manager                              */
    /*                                                                */
    /*****
    QMName[0] = 0; /* default */
    if (argc > 2)
        strcpy(QMName, argv[2]);
    MQCONN(QMName, /* queue manager          */
           &Hcon, /* connection handle          */
           &Compcode, /* completion code          */
           &Reason); /* reason code                */
    /* report reason and stop if it failed */
    if (CompCode == MQCC_FAILED)
    {
        printf("MQCONN ended with reason code %d\n", CReason);
        exit( (int)CReason );

```



```

}

/*****
/*
/* Use parameter as the name of the target queue
/*
/*
*****/
strncpy(od.ObjectName, argv[1], (size_t)MQ_Q_NAME_LENGTH);
printf("target queue is %s\n", od.ObjectName);

if (argc > 5)
{
    strncpy(od.ObjectQMgrName, argv[5], (size_t) MQ_Q_MGR_NAME_LENGTH);
    printf("target queue manager is %s\n", od.ObjectQMgrName);
}

if (argc > 6)
{
    strncpy(od.DynamicQName, argv[6], (size_t) MQ_Q_NAME_LENGTH);
    printf("dynamic queue name is %s\n", od.DynamicQName);
}

/*****
/*
/* Open the target message queue for output
/*
/*
*****/
if (argc > 3)
{
    O_options = atoi( argv[3] );
    printf("open options are %d\n", O_options);
}
else
{
    O_options = MQOO_OUTPUT          /* open queue for output      */
                | MQOO_FAIL_IF_QUIESCING /* but not if MQM stopping */
                ;                  /* = 0x2010 = 8208 decimal */
}

MQOPEN(Hcon,          /* connection handle      */
        &od,          /* object descriptor for queue */
        O_options,    /* open options           */
        &Hobj,        /* object handle          */
        &OpenCode,    /* MQOPEN completion code */
        &Reason);    /* reason code            */

/* report reason, if any; stop if failed */
if (Reason != MQRC_NONE)
{
    printf("MQOPEN ended with reason code %d\n", Reason);
}

if (OpenCode == MQCC_FAILED)
{
    printf("unable to open queue for output\n");
}

/*****
/*
/* Read lines from the file and put them to the message queue
/* Loop until null line or end of file, or there is a failure
/*
*****/
CompCode = OpenCode; /* use MQOPEN result for initial test */
fp = stdin;

memcpy(md.Format, /* character string format */
        MQFMT_STRING, (size_t)MQ_FORMAT_LENGTH);

/*****
/* These options specify that put operation should occur
/* asynchronously and the application will check the success
/* using MQSTAT at a later time.
*****/
md.Persistence = MQPER_NOT_PERSISTENT;
pmo.Options |= MQPMO_ASYNC_RESPONSE;

/*****
/* These options cause the MsgId and CorrelId to be replaced, so
/* that there is no need to reset them before each MQPUT
*****/
pmo.Options |= MQPMO_NEW_MSG_ID;

```

```

pmo.Options |= MQPMO_NEW_CORREL_ID;

while (CompCode != MQCC_FAILED)
{
    if (fgets(buffer, sizeof(buffer), fp) != NULL)
    {
        messlen = (MQLONG)strlen(buffer); /* length without null */
        if (buffer[messlen-1] == '\n') /* last char is a new-line */
        {
            buffer[messlen-1] = '\0'; /* replace new-line with null */
            --messlen; /* reduce buffer length */
        }
    }
    else messlen = 0; /* treat EOF same as null line */

    /******
    /* Put each buffer to the message queue */
    /******
    if (messlen > 0)
    {
        MQPUT(Hcon, /* connection handle */
            Hobj, /* object handle */
            &md, /* message descriptor */
            &pmo, /* default options (datagram) */
            messlen, /* message length */
            buffer, /* message buffer */
            &CompCode, /* completion code */
            &Reason); /* reason code */

        /* report reason, if any */
        if (Reason != MQRC_NONE)
        {
            printf("MQPUT ended with reason code %d\n", Reason);
        }
    }
    else /* satisfy end condition when empty line is read */
        CompCode = MQCC_FAILED;
}

/******
/* Close the target queue (if it was opened) */
/******
if (OpenCode != MQCC_FAILED)
{
    if (argc > 4)
    {
        C_options = atoi( argv[4] );
        printf("close options are %d\n", C_options);
    }
    else
    {
        C_options = MQCO_NONE; /* no close options */
    }

    MQCLOSE(Hcon, /* connection handle */
        &Hobj, /* object handle */
        C_options,
        &CompCode, /* completion code */
        &Reason); /* reason code */

    /* report reason, if any */
    if (Reason != MQRC_NONE)
    {
        printf("MQCLOSE ended with reason code %d\n", Reason);
    }
}

/******
/* Query how many asynchronous puts succeeded */
/******
MQSTAT(&Hcon, /* connection handle */
    MQSTAT_TYPE_ASYNC_ERROR, /* status type */
    &Sts, /* MQSTS structure */
    &CompCode, /* completion code */
    &Reason); /* reason code */

```

```

/* report reason, if any      */
if (Reason != MQRC_NONE)
{
    printf("MQSTAT ended with reason code %d\n", Reason);
}
else
{
    /* Display results */
    printf("Succeeded putting %d messages\n",
        sts.PutSuccessCount);
    printf("%d messages were put with a warning\n",
        sts.PutWarningCount);
    printf("Failed to put %d messages\n",
        sts.PutFailureCount);

    if(sts.CompCode == MQCC_WARNING)
    {
        printf("The first warning that occurred had reason code %d\n",
            sts.Reason);
    }
    else if(sts.CompCode == MQCC_FAILED)
    {
        printf("The first error that occurred had reason code %d\n",
            sts.Reason);
    }
}

/*****
/*
/* Disconnect from MQM if not already connected
/*
/*
/*
/*****
if (CReason != MQRC_ALREADY_CONNECTED)
{
    MQDISC(&Hcon,          /* connection handle      */
        &CompCode,       /* completion code      */
        &Reason);        /* reason code          */

    /* report reason, if any      */
    if (Reason != MQRC_NONE)
    {
        printf("MQDISC ended with reason code %d\n", Reason);
    }
}

/*****
/*
/* END OF AMQSAPT0
/*
/*
/*****
printf("Sample AMQSAPT0 end\n");
return(0);
}

```

## Exemplos COBOL

Esta coleção de tópicos é obtida dos aplicativos de amostra IBM MQ for z/OS . Eles são aplicáveis a todas as plataformas, exceto onde indicado.

### **Conectando-se a um gerenciador de filas**

Este exemplo demonstra como usar a chamadas MQCONN para conectar um programa a um gerenciador de filas em lote z/OS

Essa extração é obtida do aplicativo de amostra Browse (programa CSQ4BVA1) fornecido com IBM MQ for z/OS. Para obter os nomes e os locais dos aplicativos de amostra em outras plataformas, consulte Programas processuais de amostra (plataformas, exceto z/OS).

```

* -----*
WORKING-STORAGE SECTION.
* -----*
*   W02 - Data fields derived from the PARM field
01  W02-MQM          PIC X(48) VALUE SPACES.
*   W03 - MQM API fields
01  W03-HCONN       PIC S9(9) BINARY.
01  W03-COMPCODE    PIC S9(9) BINARY.

```

```

01 W03-REASON          PIC S9(9) BINARY.
*
*   MQV contains constants (for filling in the control
*   blocks)
*   and return codes (for testing the result of a call)
*
01 W05-MQM-CONSTANTS.
COPY CMQV SUPPRESS.
:
*   Separate into the relevant fields any data passed
*   in the PARM statement
*
UNSTRING PARM-STRING DELIMITED BY ALL ','
          INTO W02-MQM
          W02-OBJECT.
:
*   Connect to the specified queue manager.
*
CALL 'MQCONN' USING W02-MQM
                  W03-HCONN
                  W03-COMPCODE
                  W03-REASON.
*
*   Test the output of the connect call.  If the call
*   fails, print an error message showing the
*   completion code and reason code.
*
IF (W03-COMPCODE NOT = MQCC-OK) THEN
:
END-IF.
:

```

### ***desconectando-se de um gerenciador de filas***

Este exemplo demonstra como usar a chamada MQDISC para desconectar um programa de um gerenciador de fila em lote do z/OS

As variáveis usadas nessa extração de códigos são aquelas que foram configuradas no [“Conectando-se a um gerenciador de filas”](#) na página 23 Essa extração é obtida do aplicativo de amostra Browse (programa CSQ4BVA1) fornecido com IBM MQ for z/OS. Para obter os nomes e os locais dos aplicativos de amostra em outras plataformas, consulte [Programas processuais de amostra \(plataformas, exceto z/OS\)](#).

```

:
*
*   Disconnect from the queue manager
*
CALL 'MQDISC' USING W03-HCONN
                  W03-COMPCODE
                  W03-REASON.
*
*   Test the output of the disconnect call.  If the
*   call fails, print an error message showing the
*   completion code and reason code.
*
IF (W03-COMPCODE NOT = MQCC-OK) THEN
:
END-IF.
:

```

### ***Criando uma fila dinâmica***

Este exemplo demonstra como usar a chamada MQOPEN para criar uma fila dinâmica..

Essa extração é obtida do aplicativo de amostra Credit Check (programa CSQ4CVB1) fornecido com IBM MQ for z/OS. Para obter os nomes e os locais dos aplicativos de amostra em outras plataformas, consulte [Programas processuais de amostra \(plataformas, exceto z/OS\)](#).

```

:
* -----*
WORKING-STORAGE SECTION.
* -----*
*
*   W02 - Queues processed in this program
*

```

```

01 W02-MODEL-QNAME      PIC X(48) VALUE
   'CSQ4SAMP.B1.MODEL      '
01 W02-NAME-PREFIX     PIC X(48) VALUE
   'CSQ4SAMP.B1.*         '
01 W02-TEMPORARY-Q     PIC X(48).
*
*   W03 - MQM API fields
*
01 W03-HCONN          PIC S9(9) BINARY VALUE ZERO.
01 W03-OPTIONS       PIC S9(9) BINARY.
01 W03-HOBJ          PIC S9(9) BINARY.
01 W03-COMPCODE      PIC S9(9) BINARY.
01 W03-REASON        PIC S9(9) BINARY.
*
*   API control blocks
*
01 MQM-OBJECT-DESCRIPTOR.
   COPY CMQODV.
*
*   CMQV contains constants (for setting or testing
*   field values) and return codes (for testing the
*   result of a call)
*
01 MQM-CONSTANTS.
   COPY CMQV SUPPRESS.
* -----*
PROCEDURE DIVISION.
* -----*
:
* -----*
OPEN-TEMP-RESPONSE-QUEUE SECTION.
* -----*

```

```

*
*   This section creates a temporary dynamic queue
*   using a model queue
*
* -----*
*
*   Change three fields in the Object Descriptor (MQOD)
*   control block. (MQODV initializes the other fields)
*
   MOVE MQOT-Q          TO MQOD-OBJECTTYPE.
   MOVE W02-MODEL-QNAME TO MQOD-OBJECTNAME.
   MOVE W02-NAME-PREFIX TO MQOD-DYNAMICQNAME.
*
   COMPUTE W03-OPTIONS = MQOD-INPUT-EXCLUSIVE.
*
   CALL 'MQOPEN' USING W03-HCONN
                      MQOD
                      W03-OPTIONS
                      W03-HOBJ-MODEL
                      W03-COMPCODE
                      W03-REASON.
*
   IF W03-COMPCODE NOT = MQCC-OK
       MOVE 'MQOPEN'      TO M01-MSG4-OPERATION
       MOVE W03-COMPCODE TO M01-MSG4-COMPCODE
       MOVE W03-REASON   TO M01-MSG4-REASON
       MOVE M01-MESSAGE-4 TO M00-MESSAGE
   ELSE
       MOVE MQOD-OBJECTNAME TO W02-TEMPORARY-Q
   END-IF.
*
OPEN-TEMP-RESPONSE-QUEUE-EXIT.
*
*   Return to performing section.
*
EXIT.
EJECT
*

```

### ***Abrindo uma fila existente***

Este exemplo demonstra como usar a chamada MQOPEN para abrir uma fila existente..

Essa extração é obtida do aplicativo de amostra Browse (programa CSQ4BVA1) fornecido com IBM MQ for z/OS. Para obter os nomes e os locais dos aplicativos de amostra em outras plataformas, consulte Programas processuais de amostra (plataformas, exceto z/OS).

```

:
* -----*
WORKING-STORAGE SECTION.
* -----*
*
*   W01 - Fields derived from the command area input
*
01 W01-OBJECT          PIC X(48).
*
*   W02 - MQM API fields
*
01 W02-HCONN          PIC S9(9) BINARY VALUE ZERO.
01 W02-OPTIONS        PIC S9(9) BINARY.
01 W02-HOBJ           PIC S9(9) BINARY.
01 W02-COMPCODE       PIC S9(9) BINARY.
01 W02-REASON         PIC S9(9) BINARY.
*
*   CMQODV defines the object descriptor (MQOD)
*
01 MQM-OBJECT-DESCRIPTOR.
   COPY CMQODV.
*
* CMQV contains constants (for setting or testing
* field values) and return codes (for testing the
* result of a call)
*
01 MQM-CONSTANTS.
   COPY CMQV SUPPRESS.
* -----*
E-OPEN-QUEUE SECTION.
* -----*
*
* This section opens the queue
*
*   Initialize the Object Descriptor (MQOD) control
*   block
*   (The copy file initializes the remaining fields.)
*
MOVE MQOT-Q           TO MQOD-OBJECTTYPE.
MOVE W01-OBJECT       TO MQOD-OBJECTNAME.
*
*   Initialize W02-OPTIONS to open the queue for both
*   inquiring about and setting attributes
*
COMPUTE W02-OPTIONS = MQ00-INQUIRE + MQ00-SET.

```

```

*
*   Open the queue
*
CALL 'MQOPEN' USING W02-HCONN
                  MQOD
                  W02-OPTIONS
                  W02-HOBJ
                  W02-COMPCODE
                  W02-REASON.
*
*   Test the output from the open
*
*   If the completion code is not OK, display a
*   separate error message for each of the following
*   errors:
*
*   Q-MGR-NOT-AVAILABLE - MQM is not available
*   CONNECTION-BROKEN  - MQM is no longer connected to CICS
*   UNKNOWN-OBJECT-NAME - The queue does not exist
*   NOT-AUTHORIZED     - The user is not authorized to open
*                       the queue
*
*   For any other error, display an error message
*   showing the completion and reason codes
*
IF W02-COMPCODE NOT = MQCC-OK
EVALUATE TRUE

```

```

*
*   WHEN W02-REASON = MQRC-Q-MGR-NOT-AVAILABLE
*       MOVE M01-MESSAGE-6 TO M00-MESSAGE
*
*   WHEN W02-REASON = MQRC-CONNECTION-BROKEN
*       MOVE M01-MESSAGE-6 TO M00-MESSAGE
*
*   WHEN W02-REASON = MQRC-UNKNOWN-OBJECT-NAME
*       MOVE M01-MESSAGE-2 TO M00-MESSAGE
*
*   WHEN W02-REASON = MQRC-NOT-AUTHORIZED
*       MOVE M01-MESSAGE-3 TO M00-MESSAGE
*
*   WHEN OTHER
*       MOVE 'MQOPEN'      TO M01-MSG4-OPERATION
*       MOVE W02-COMPCODE TO M01-MSG4-COMPCODE
*       MOVE W02-REASON   TO M01-MSG4-REASON
*       MOVE M01-MESSAGE-4 TO M00-MESSAGE
*   END-EVALUATE
* END-IF.
* E-EXIT.
*
*   Return to performing section
*
*   EXIT.
*   EJECT

```

## ***fechar uma fila***

Este exemplo demonstra como usar a chamada MQCLOSE

As variáveis usadas nessa extração de códigos são aquelas que foram configuradas no [“Conectando-se a um gerenciador de filas”](#) na página 23 Essa extração é obtida do aplicativo de amostra Browse (programa CSQ4BVA1) fornecido com IBM MQ for z/OS. Para obter os nomes e os locais dos aplicativos de amostra em outras plataformas, consulte [Programas processuais de amostra \(plataformas, exceto z/OS\)](#).

```

:
*
*   Close the queue
*
*   MOVE MQCO-NONE TO W03-OPTIONS.
*
*   CALL 'MQCLOSE' USING W03-HCONN
*                       W03-HOBJ
*                       W03-OPTIONS
*                       W03-COMPCODE
*                       W03-REASON.
*
*   Test the output of the MQCLOSE call.  If the call
*   fails, print an error message showing the
*   completion code and reason code.
*
*   IF (W03-COMPCODE NOT = MQCC-OK) THEN
*       MOVE 'CLOSE'      TO W04-MSG4-TYPE
*       MOVE W03-COMPCODE TO W04-MSG4-COMPCODE
*       MOVE W03-REASON   TO W04-MSG4-REASON
*       MOVE W04-MESSAGE-4 TO W00-PRINT-DATA
*       PERFORM PRINT-LINE
*       MOVE W06-CSQ4-ERROR TO W00-RETURN-CODE
*   END-IF.
*

```

## ***Colocando uma mensagem usando MQPUT***

Este exemplo demonstra como usar a chamada MQPUT usando o contexto

Essa extração é obtida do aplicativo de amostra Credit Check (programa CSQ4CVB1) fornecido com IBM MQ for z/OS. Para obter os nomes e os locais dos aplicativos de amostra em outras plataformas, consulte [Programas processuais de amostra \(plataformas, exceto z/OS\)](#).

```

:
* -----*
* WORKING-STORAGE SECTION.
* -----*
*

```

```

*   W02 - Queues processed in this program
*
01  W02-TEMPORARY-Q           PIC X(48).
*
*   W03 - MQM API fields
*
01  W03-HCONN                 PIC S9(9) BINARY VALUE ZERO.
01  W03-HOBJ-INQUIRY         PIC S9(9) BINARY.
01  W03-OPTIONS              PIC S9(9) BINARY.
01  W03-BUFFLEN              PIC S9(9) BINARY.
01  W03-COMPCODE             PIC S9(9) BINARY.
01  W03-REASON               PIC S9(9) BINARY.
*
01  W03-PUT-BUFFER.
*
05  W03-CSQ4BIIM.
    COPY CSQ4VB1.
*
*   API control blocks
*
01  MQM-MESSAGE-DESCRIPTOR.
    COPY CMQMDV.
01  MQM-PUT-MESSAGE-OPTIONS.
    COPY CMQPMOV.
*
*   MQV contains constants (for filling in the
*   control blocks) and return codes (for testing
*   the result of a call).
*
01  MQM-CONSTANTS.
    COPY CMQV SUPPRESS.
* -----*
PROCEDURE DIVISION.
* -----*
:
*   Open queue and build message.
:

```

```

*
* Set the message descriptor and put-message options to
* the values required to create the message.
* Set the length of the message.
*
MOVE MQMT-REQUEST           TO MQMD-MSGTYPE.
MOVE MQCI-NONE              TO MQMD-CORRELID.
MOVE MQMI-NONE              TO MQMD-MSGID.
MOVE W02-TEMPORARY-Q       TO MQMD-REPLYTOQ.
MOVE SPACES                 TO MQMD-REPLYTOQMGR.
MOVE 5                      TO MQMD-PRIORITY.
MOVE MQPER-NOT-PERSISTENT  TO MQMD-PERSISTENCE.
COMPUTE MQPMO-OPTIONS      = MQPMO-NO-SYNCPOINT +
                           MQPMO-DEFAULT-CONTEXT.
MOVE LENGTH OF CSQ4BIIM-MSG TO W03-BUFFLEN.
*
CALL 'MQPUT' USING W03-HCONN
                  W03-HOBJ-INQUIRY
                  MQMD
                  MQPMO
                  W03-BUFFLEN
                  W03-PUT-BUFFER
                  W03-COMPCODE
                  W03-REASON.
IF W03-COMPCODE NOT = MQCC-OK
:
END-IF.

```

### ***Colocando uma mensagem usando MQPUT1***

Este exemplo demonstra como usar a chamada MQPUT1

Essa extração é obtida do aplicativo de amostra Credit Check (programa CSQ4CVB5) fornecido com IBM MQ for z/OS. Para obter os nomes e os locais dos aplicativos de amostra em outras plataformas, consulte [Programas processuais de amostra \(plataformas, exceto z/OS\)](#).

```

:
* -----*

```



```

WORKING-STORAGE SECTION.
* -----*
*
*   W03 - MQM API fields
*
01 W03-HCONN          PIC S9(9) BINARY VALUE ZERO.
01 W03-OPTIONS       PIC S9(9) BINARY.
01 W03-COMPCODE      PIC S9(9) BINARY.
01 W03-REASON        PIC S9(9) BINARY.
01 W03-BUFFLEN       PIC S9(9) BINARY.
*
01 W03-PUT-BUFFER.
   05 W03-CSQ4BQRM.
   COPY CSQ4VB4.

*
*   API control blocks
*
01 MQM-OBJECT-DESCRIPTOR.
   COPY CMQODV.
01 MQM-MESSAGE-DESCRIPTOR.
   COPY CMQMDV.
01 MQM-PUT-MESSAGE-OPTIONS.
   COPY CMQPMOV.
*
* CMQV contains constants (for filling in the
* control blocks) and return codes (for testing
* the result of a call).
*
01 MQM-MQV.
   COPY CMQV SUPPRESS.
* -----*
PROCEDURE DIVISION.
* -----*
:
*   Get the request message.
:
* -----*
PROCESS-QUERY SECTION.
* -----*
:
*   Build the reply message.
:
*
* Set the object descriptor, message descriptor and
* put-message options to the values required to create
* the message.
* Set the length of the message.
*
MOVE MQMD-REPLYTOQ    TO MQOD-OBJECTNAME.
MOVE MQMD-REPLYTOQMGR TO MQOD-OBJECTQMGRNAME.
MOVE MQMT-REPLY       TO MQMD-MSGTYPE.
MOVE SPACES           TO MQMD-REPLYTOQ.
MOVE SPACES           TO MQMD-REPLYTOQMGR.
MOVE LOW-VALUES       TO MQMD-MSGID.
COMPUTE MQPMO-OPTIONS = MQPMO-SYNCPPOINT +
                      MQPMO-PASS-IDENTITY-CONTEXT.
MOVE W03-HOBJ-CHECKQ  TO MQPMO-CONTEXT.
MOVE LENGTH OF CSQ4BQRM-MSG TO W03-BUFFLEN.
*
CALL 'MQPUT1' USING W03-HCONN
                  MQOD
                  MQMD
                  MQPMO
                  W03-BUFFLEN
                  W03-PUT-BUFFER
                  W03-COMPCODE
                  W03-REASON.
IF W03-COMPCODE NOT = MQCC-OK
  MOVE 'MQPUT1'      TO M02-OPERATION
  MOVE MQOD-OBJECTNAME TO M02-OBJECTNAME
  PERFORM RECORD-CALL-ERROR
  PERFORM FORWARD-MSG-TO-DLQ
END-IF.
*

```

### ***Obtendo uma mensagem***

Este exemplo demonstra como usar a chamada MQGET para remover uma mensagem de uma fila.

Essa extração é obtida do aplicativo de amostra Credit Check (programa CSQ4CVB1) fornecido com IBM MQ for z/OS. Para obter os nomes e os locais dos aplicativos de amostra em outras plataformas, consulte Programas processuais de amostra (plataformas, exceto z/OS).

```

:
* -----*
WORKING-STORAGE SECTION.
* -----*
*
*   W03 - MQM API fields
*
01 W03-HCONN          PIC S9(9) BINARY VALUE ZERO.
01 W03-HOBJ-RESPONSE PIC S9(9) BINARY.
01 W03-OPTIONS       PIC S9(9) BINARY.
01 W03-BUFFLEN       PIC S9(9) BINARY.
01 W03-DATALEN       PIC S9(9) BINARY.
01 W03-COMPCODE      PIC S9(9) BINARY.
01 W03-REASON        PIC S9(9) BINARY.
*
01 W03-GET-BUFFER.
   05 W03-CSQ4BAM.
   COPY CSQ4VB2.
*
*   API control blocks
*
01 MQM-MESSAGE-DESCRIPTOR.
   COPY CMQMDV.
01 MQM-GET-MESSAGE-OPTIONS.
   COPY CMQGMV.
*
*   MQV contains constants (for filling in the
*   control blocks) and return codes (for testing
*   the result of a call).
*
01 MQM-CONSTANTS.
   COPY CMQV SUPPRESS.
* -----*
A-MAIN SECTION.
* -----*
:
*   Open response queue.
:
* -----*
PROCESS-RESPONSE-SCREEN SECTION.
* -----*
*
*   This section gets a message from the response queue.
*
*   When a correct response is received, it is
*   transferred to the map for display; otherwise
*   an error message is built.
*
* -----*

```

```

*
*   Set get-message options
*
*   COMPUTE MQGMO-OPTIONS = MQGMO-SYNCPOINT +
*                           MQGMO-ACCEPT-TRUNCATED-MSG +
*                           MQGMO-NO-WAIT.
*
*   Set msgid and correlid in MQMD to nulls so that any
*   message will qualify.
*   Set length to available buffer length.
*
*   MOVE MQMI-NONE TO MQMD-MSGID.
*   MOVE MQCI-NONE TO MQMD-CORRELID.
*   MOVE LENGTH OF W03-GET-BUFFER TO W03-BUFFLEN.
*
*   CALL 'MQGET' USING W03-HCONN
*                       W03-HOBJ-RESPONSE
*                       MQMD
*                       MQGMO
*                       W03-BUFFLEN
*                       W03-GET-BUFFER
*                       W03-DATALEN
*                       W03-COMPCODE

```

```

                                W03-REASON.
EVALUATE TRUE
  WHEN W03-COMPCODE NOT = MQCC-FAILED
  :
*   : Process the message
  :
  WHEN (W03-COMPCODE = MQCC-FAILED AND
        W03-REASON = MQRC-NO-MSG-AVAILABLE)
    MOVE M01-MESSAGE-9 TO M00-MESSAGE
    PERFORM CLEAR-RESPONSE-SCREEN
*
  WHEN OTHER
    MOVE 'MQGET '      TO M01-MSG4-OPERATION
    MOVE W03-COMPCODE TO M01-MSG4-COMPCODE
    MOVE W03-REASON   TO M01-MSG4-REASON
    MOVE M01-MESSAGE-4 TO M00-MESSAGE
    PERFORM CLEAR-RESPONSE-SCREEN
END-EVALUATE.

```

### ***Obtendo uma mensagem usando a opção de espera***

Este exemplo demonstra como usar a chamada MQGET com a opção de espera e aceitar mensagens truncadas.

Essa extração é obtida do aplicativo de amostra Credit Check (programa CSQ4CVB5) fornecido com IBM MQ for z/OS. Para obter os nomes e os locais dos aplicativos de amostra em outras plataformas, consulte [Programas processuais de amostra \(plataformas, exceto z/OS\)](#).

```

:
* -----*
* WORKING-STORAGE SECTION.
* -----*
*
*   W00 - General work fields
*
*01 W00-WAIT-INTERVAL   PIC S9(09) BINARY VALUE 30000.
*
*   W03 - MQM API fields
*
*01 W03-HCONN          PIC S9(9) BINARY VALUE ZERO.
*01 W03-OPTIONS        PIC S9(9) BINARY.
*01 W03-HOBJ-CHECKQ    PIC S9(9) BINARY.
*01 W03-COMPCODE       PIC S9(9) BINARY.
*01 W03-REASON         PIC S9(9) BINARY.
*01 W03-DATALEN        PIC S9(9) BINARY.
*01 W03-BUFFLEN        PIC S9(9) BINARY.
*
*01 W03-MSG-BUFFER.
*   05 W03-CSQ4BCAQ.
*   COPY CSQ4VB3.
*
*   API control blocks
*
*01 MQM-MESSAGE-DESCRIPTOR.
*   COPY CMQMDV.
*01 MQM-GET-MESSAGE-OPTIONS.
*   COPY CMQGMV.
*
*   CMQV contains constants (for filling in the
*   control blocks) and return codes (for testing
*   the result of a call).
*
*01 MQM-MQV.
*   COPY CMQV SUPPRESS.
* -----*
* PROCEDURE DIVISION.
* -----*
:
*   Open input queue.
:

```

```

*
*   Get and process messages.
*
*   COMPUTE MQGMO-OPTIONS = MQGMO-WAIT +
*                           MQGMO-ACCEPT-TRUNCATED-MSG +

```

```

                                MQGMO-SYNCPOINT.
MOVE LENGTH OF W03-MSG-BUFFER TO W03-BUFFLEN.
MOVE W00-WAIT-INTERVAL TO MQGMO-WAITINTERVAL.
MOVE MQMI-NONE TO MQMD-MSGID.
MOVE MQCI-NONE TO MQMD-CORRELID.
*
*   Make the first MQGET call outside the loop.
*
CALL 'MQGET' USING W03-HCONN
                  W03-HOBJ-CHECKQ
                  MQMD
                  MQGMO
                  W03-BUFFLEN
                  W03-MSG-BUFFER
                  W03-DATALEN
                  W03-COMPCODE
                  W03-REASON.
*
*   Test the output of the MQGET call using the
*   PERFORM loop that follows.
*
*   Perform whilst no failure occurs
*   - process this message
*   - reset the call parameters
*   - get another message
*   End-perform
*
*
*   Test the output of the MQGET call.  If the call
*   fails, send an error message showing the
*   completion code and reason code, unless the
*   completion code is NO-MSG-AVAILABLE.
*
IF (W03-COMPCODE NOT = MQCC-FAILED) OR
   (W03-REASON NOT = MQRC-NO-MSG-AVAILABLE)
  MOVE 'MQGET '          TO M02-OPERATION
  MOVE MQ0D-OBJECTNAME  TO M02-OBJECTNAME
  PERFORM RECORD-CALL-ERROR
END-IF.
:

```

### ***Obtendo uma mensagem usando sinalização***

Este exemplo demonstra como usar a chamada MQGET com sinalização.. Essa extração é obtida do aplicativo de amostra Credit Check (programa CSQ4CVB2) fornecido com IBM MQ for z/OS.

*A sinalização está disponível apenas com o IBM MQ for z/OS*

```

:
* -----*
WORKING-STORAGE SECTION.
* -----*
*
*   W00 - General work fields
*
01 W00-WAIT-INTERVAL    PIC S9(09) BINARY VALUE 30000.
*
*   W03 - MQM API fields
*
01 W03-HCONN           PIC S9(9) BINARY VALUE ZERO.
01 W03-HOBJ-REPLYQ    PIC S9(9) BINARY.
01 W03-COMPCODE        PIC S9(9) BINARY.
01 W03-REASON          PIC S9(9) BINARY.
01 W03-DATALEN         PIC S9(9) BINARY.
01 W03-BUFFLEN        PIC S9(9) BINARY.
:
01 W03-GET-BUFFER.
05 W03-CSQ4BQRM.
COPY CSQ4VB4.
*
05 W03-CSQ4BIIM REDEFINES W03-CSQ4BQRM.
COPY CSQ4VB1.
*
05 W03-CSQ4BPGM REDEFINES W03-CSQ4BIIM.
COPY CSQ4VB5.
:

```

```

*   API control blocks
*
01  MQM-MESSAGE-DESCRIPTOR.
    COPY CMQMDV.
01  MQM-GET-MESSAGE-OPTIONS.
    COPY CMQGMV.
    :
*   MQV contains constants (for filling in the
*   control blocks) and return codes (for testing
*   the result of a call).
*
01  MQM-MQV.
    COPY CMQV SUPPRESS.
* -----*
LINKAGE SECTION.
* -----*
01  L01-ECB-ADDR-LIST.
     05  L01-ECB-ADDR1          POINTER.
     05  L01-ECB-ADDR2          POINTER.

*
01  L02-ECBS.
     05  L02-INQUIRY-ECB1       PIC S9(09) BINARY.
     05  L02-REPLY-ECB2        PIC S9(09) BINARY.
01  REDEFINES L02-ECBS.
     05                                     PIC X(02).
     05  L02-INQUIRY-ECB1-CC   PIC S9(04) BINARY.
     05                                     PIC X(02).
     05  L02-REPLY-ECB2-CC     PIC S9(04) BINARY.

*
* -----*
PROCEDURE DIVISION.
* -----*
:
* Initialize variables, open queues, set signal on
* inquiry queue.
:
* -----*
PROCESS-SIGNAL-ACCEPTED SECTION.
* -----*
* This section gets a message with signal.  If a
* message is received, process it.  If the signal
* is set or is already set, the program goes into
* an operating system wait.
* Otherwise an error is reported and call error set.
* -----*
*
PERFORM REPLYQ-GETSIGNAL.
*
EVALUATE TRUE
  WHEN (W03-COMPCODE = MQCC-OK AND
        W03-REASON = MQRC-NONE)
    PERFORM PROCESS-REPLYQ-MESSAGE
*
  WHEN (W03-COMPCODE = MQCC-WARNING AND
        W03-REASON = MQRC-SIGNAL-REQUEST-ACCEPTED)
    OR
    (W03-COMPCODE = MQCC-FAILED AND
     W03-REASON = MQRC-SIGNAL-OUTSTANDING)
    PERFORM EXTERNAL-WAIT
*
  WHEN OTHER
    MOVE 'MQGET SIGNAL' TO M02-OPERATION
    MOVE MQOD-OBJECTNAME TO M02-OBJECTNAME
    PERFORM RECORD-CALL-ERROR
    MOVE W06-CALL-ERROR TO W06-CALL-STATUS
END-EVALUATE.
*
PROCESS-SIGNAL-ACCEPTED-EXIT.
* Return to performing section
EXIT.
EJECT
*
* -----*
EXTERNAL-WAIT SECTION.
* -----*
* This section performs an external CICS wait on two
*

```

```

* ECBs until at least one is posted. It then calls      *
* the sections to handle the posted ECB.                *
* -----*
* EXEC CICS WAIT EXTERNAL                                *
*   ECBLIST(W04-ECB-ADDR-LIST-PTR)                      *
*   NUMEVENTS(2)                                         *
* END-EXEC.                                              *
*
* At least one ECB must have been posted to get to this *
* point. Test which ECB has been posted and perform     *
* the appropriate section.                              *
*
* IF L02-INQUIRY-ECB1 NOT = 0                            *
*   PERFORM TEST-INQUIRYQ-ECB                          *
* ELSE                                                  *
*   PERFORM TEST-REPLYQ-ECB                            *
* END-IF.
*
* EXTERNAL-WAIT-EXIT.
*
* Return to performing section.
*
* EXIT.
* EJECT
* :
* -----*
* REPLYQ-GETSIGNAL SECTION.
* -----*
*
* This section performs an MQGET call (in syncpoint with *
* signal) on the reply queue. The signal field in the   *
* MQGMO is set to the address of the ECB.              *
* Response handling is done by the performing section.  *
*
* -----*
*
* COMPUTE MQGMO-OPTIONS = MQGMO-SYNCPOINT +
*                        MQGMO-SET-SIGNAL.
* MOVE W00-WAIT-INTERVAL TO MQGMO-WAITINTERVAL.
* MOVE LENGTH OF W03-GET-BUFFER TO W03-BUFFLEN.
*
* MOVE ZEROS TO L02-REPLY-ECB2.
* SET MQGMO-SIGNAL1 TO ADDRESS OF L02-REPLY-ECB2.
*
*
*
* Set msgid and correlid to nulls so that any message
* will qualify.
*
* MOVE MQMI-NONE TO MQMD-MSGID.
* MOVE MQCI-NONE TO MQMD-CORRELID.
*
* CALL 'MQGET' USING W03-HCONN
*                   W03-HOBJ-REPLYQ
*                   MQMD
*                   MQGMO
*                   W03-BUFFLEN
*                   W03-GET-BUFFER
*                   W03-DATALEN
*                   W03-COMPCODE
*                   W03-REASON.
*
* REPLYQ-GETSIGNAL-EXIT.
*
* Return to performing section.
*
* EXIT.
* EJECT
*
* :

```

### ***Consultando sobre os atributos de um objeto***

Este exemplo demonstra como usar a chamada MQINQ para consultar sobre os atributos de uma fila

Essa extração é obtida do aplicativo de amostra de Atributos da Fila (programa CSQ4CVC1) fornecido com IBM MQ for z/OS. Para obter os nomes e os locais dos aplicativos de amostra em outras plataformas, consulte Programas processuais de amostra (plataformas, exceto z/OS).

```

:
* -----*
WORKING-STORAGE SECTION.
* -----*
*
*   W02 - MQM API fields
*
01 W02-SELECTORCOUNT    PIC S9(9) BINARY VALUE 2.
01 W02-INTATTRCOUNT    PIC S9(9) BINARY VALUE 2.
01 W02-CHARATTRLENGTH   PIC S9(9) BINARY VALUE ZERO.
01 W02-CHARATTRS        PIC X    VALUE LOW-VALUES.
01 W02-HCONN            PIC S9(9) BINARY VALUE ZERO.
01 W02-HOBJ             PIC S9(9) BINARY.
01 W02-COMPCODE         PIC S9(9) BINARY.
01 W02-REASON           PIC S9(9) BINARY.
01 W02-SELECTORS-TABLE.
   05 W02-SELECTORS     PIC S9(9) BINARY OCCURS 2 TIMES
01 W02-INTATTRS-TABLE.
   05 W02-INTATTRS     PIC S9(9) BINARY OCCURS 2 TIMES
*
*   CMQODV defines the object descriptor (MQOD).
*
01 MQM-OBJECT-DESCRIPTOR.
   COPY CMQODV.
*
*   CMQV contains constants (for setting or testing field
*   values) and return codes (for testing the result of a
*   call).
*
01 MQM-CONSTANTS.
   COPY CMQV SUPPRESS.
* -----*
PROCEDURE DIVISION.
* -----*
*
*   Get the queue name and open the queue.
*
:
*
*   Initialize the variables for the inquiry call:
*   - Set W02-SELECTORS-TABLE to the attributes whose
*   status is required
*   - All other variables are already set
*
MOVE MQIA-INHIBIT-GET TO W02-SELECTORS(1).
MOVE MQIA-INHIBIT-PUT TO W02-SELECTORS(2).

```

```

*
*   Inquire about the attributes.
*
CALL 'MQINQ' USING W02-HCONN,
                  W02-HOBJ,
                  W02-SELECTORCOUNT,
                  W02-SELECTORS-TABLE,
                  W02-INTATTRCOUNT,
                  W02-INTATTRS-TABLE,
                  W02-CHARATTRLENGTH,
                  W02-CHARATTRS,
                  W02-COMPCODE,
                  W02-REASON.
*
*   Test the output from the inquiry:
*
*   - If the completion code is not OK, display an error
*   message showing the completion and reason codes
*
*   - Otherwise, move the correct attribute status into
*   the relevant screen map fields
*
IF W02-COMPCODE NOT = MQCC-OK
  MOVE 'MQINQ'          TO M01-MSG4-OPERATION
  MOVE W02-COMPCODE     TO M01-MSG4-COMPCODE
  MOVE W02-REASON      TO M01-MSG4-REASON

```

```

        MOVE M01-MESSAGE-4 TO M00-MESSAGE
*
*   ELSE
*       Process the changes.
*       :
*       END-IF.
*       :

```

## Configurando os atributos de uma fila

Este exemplo demonstra como usar a chamada MQSET para alterar os atributos de uma fila

Essa extração é obtida do aplicativo de amostra de Atributos da Fila (programa CSQ4CVC1) fornecido com IBM MQ for z/OS. Para os nomes e locais dos aplicativos de amostra em outras plataformas, consulte Programas processuais de amostra (plataformas, exceto z/OS)

```

:
* -----*
* WORKING-STORAGE SECTION.
* -----*
*
*   W02 - MQM API fields
*
*01 W02-SELECTORCOUNT    PIC S9(9) BINARY VALUE 2.
*01 W02-INTATTRCOUNT    PIC S9(9) BINARY VALUE 2.
*01 W02-CHARATTRLENGTH   PIC S9(9) BINARY VALUE ZERO.
*01 W02-CHARATTRS        PIC X      VALUE LOW-VALUES.
*01 W02-HCONN            PIC S9(9) BINARY VALUE ZERO.
*01 W02-HOBJ             PIC S9(9) BINARY.
*01 W02-COMPCODE         PIC S9(9) BINARY.
*01 W02-REASON           PIC S9(9) BINARY.
*01 W02-SELECTORS-TABLE.
*   05 W02-SELECTORS     PIC S9(9) BINARY OCCURS 2 TIMES.
*01 W02-INTATTRS-TABLE.
*   05 W02-INTATTRS     PIC S9(9) BINARY OCCURS 2 TIMES.
*
*   CMQODV defines the object descriptor (MQOD).
*
*01 MQM-OBJECT-DESCRIPTOR.
*   COPY CMQODV.
*
*   CMQV contains constants (for setting or testing
*   field values) and return codes (for testing the
*   result of a call).
*
*01 MQM-CONSTANTS.
*   COPY CMQV SUPPRESS.
* -----*
* PROCEDURE DIVISION.
* -----*

```

```

*
*   Get the queue name and open the queue.
*
*   :
*
*   Initialize the variables required for the set call:
*   - Set W02-SELECTORS-TABLE to the attributes to be set
*   - Set W02-INTATTRS-TABLE to the required status
*   - All other variables are already set
*
*   MOVE MQIA-INHIBIT-GET    TO W02-SELECTORS(1).
*   MOVE MQIA-INHIBIT-PUT    TO W02-SELECTORS(2).
*   MOVE MQQA-GET-INHIBITED TO W02-INTATTRS(1).
*   MOVE MQQA-PUT-INHIBITED TO W02-INTATTRS(2).
*
*   Set the attributes.
*
*   CALL 'MQSET' USING W02-HCONN,
*                     W02-HOBJ,
*                     W02-SELECTORCOUNT,
*                     W02-SELECTORS-TABLE,
*                     W02-INTATTRCOUNT,
*                     W02-INTATTRS-TABLE,
*                     W02-CHARATTRLENGTH,

```



```

                                W02-CHARATTRS,
                                W02-COMPCODE,
                                W02-REASON.
*
* Test the output from the call:
*
* - If the completion code is not OK, display an error
*   message showing the completion and reason codes
*
* - Otherwise, move 'INHIBITED' into the relevant
*   screen map fields
*
IF W02-COMPCODE NOT = MQCC-OK
    MOVE 'MQSET'          TO M01-MSG4-OPERATION
    MOVE W02-COMPCODE     TO M01-MSG4-COMPCODE
    MOVE W02-REASON      TO M01-MSG4-REASON
    MOVE M01-MESSAGE-4   TO M00-MESSAGE
ELSE
*
*       Process the changes.
*
:
END-IF.

```

## System/390 exemplos de linguagem assembler

Essa coleção de tópicos é principalmente obtida dos aplicativos de amostra do IBM MQ for z/OS .

### ***Conectando-se a um gerenciador de filas***

Este exemplo demonstra como usar a chamadas MQCONN para conectar um programa a um gerenciador de filas em lote z/OS

Essa extração é obtida do programa de amostra Browse (CSQ4BAA1) fornecido com IBM MQ for z/OS.

```

:
WORKAREA DSECT
*
PARMLIST CALL ,(0,0,0,0,0,0,0,0,0,0,0,0),VL,MF=L
*
COMPCODE DS    F           Completion code
REASON   DS    F           Reason code
HCONN   DS    F           Connection handle
          ORG
PARMADDR DS    F           Address of parm field
PARMLEN DS    H           Length of parm field
*
MQMNAME DS    CL48         Queue manager name
*
*****
* SECTION NAME : MAINPARM *
*****
MAINPARM DS    0H
          MVI   MQMNAME,X'40'
          MVC   MQMNAME+1(L'MQMNAME-1),MQMNAME
*
* Space out first byte and initialize
*
*
* Code to address and verify parameters passed omitted
*
*
PARM1MVE DS    0H
          SR    R1,R3           Length of data
          LA   R4,MQMNAME       Address for target
          BCTR R1,R0           Reduce for execute
          EX   R1,MOVEPARM       Move the data
*
*****
* EXECUTES *
*****
MOVEPARM MVC    0(*-*,R4),0(R3)
*
          EJECT

```

```

*****
* SECTION NAME : MAINCONN *
*****
*
*
MAINCONN DS 0H
XC HCONN,HCONN Null connection handle
*
CALL MQCONN, X
(MQMNAME, X
HCONN, X
COMPCODE, X
REASON), X
MF=(E,PARMLIST),VL
*
LA R0,MQCC_OK Expected compcode
C R0,COMPCODE As expected?
BER R6 Yes .. return to caller
*
MVC INF4_TYP,=CL10'CONNECT '
BAL R7,ERRCODE Translate error
LA R0,8 Set exit code
ST R0,EXITCODE to 8
B ENDPROG End the program
*

```

### ***desconectando-se de um gerenciador de filas***

Este exemplo demonstra como usar a chamada MQDISC para desconectar um programa de um gerenciador de fila em lote do z/OS

Essa extração não é obtida dos aplicativos de amostra fornecidos com o IBM MQ.

```

:
*
* ISSUE MQI DISC REQUEST USING REENTRANT FORM
* OF CALL MACRO
*
* HCONN WAS SET BY A PREVIOUS MQCONN REQUEST
* R5 = WORK REGISTER
*
DISC DS 0H
CALL MQDISC, X
(HCONN, X
COMPCODE, X
REASON), X
VL,MF=(E,CALLST)
*
LA R5,MQCC_OK
C R5,COMPCODE
BNE BADCALL
:

```

```

BADCALL DS 0H
:
*
* CONSTANTS
*
* CMQA
*
* WORKING STORAGE (RE-ENTRANT)
*
WEG3 DSECT
*
CALLST CALL ,(0,0,0,0,0,0,0,0,0,0,0,0),VL,MF=L
*
HCONN DS F
COMPCODE DS F
REASON DS F
*
*
LEG3 EQU *-WKEG3
END

```

## ***Criando uma fila dinâmica***

Este exemplo demonstra como usar a chamada MQOPEN para criar uma fila dinâmica..

Essa extração não é obtida dos aplicativos de amostra fornecidos com o IBM MQ.

```
:
*
*   R5 = WORK REGISTER.
*
OPEN    DS    0H
*
*   MVC  WOD_AREA,MQOD_AREA INITIALIZE WORKING VERSION OF
*           MQOD WITH DEFAULTS
*   MVC  WOD_OBJECTNAME,MOD_Q   COPY IN THE MODEL Q NAME
*   MVC  WOD_DYNAMICQNAME,DYN_Q COPY IN THE DYNAMIC Q NAME
*   L    R5,=AL4(MQOO_OUTPUT)   OPEN FOR OUTPUT AND
*   A    R5,=AL4(MQOO_INQUIRE) INQUIRE
*   ST   R5,OPTIONS

*
* ISSUE MQI OPEN REQUEST USING REentrant
* FORM OF CALL MACRO
*
*   CALL MQOPEN,                X
*           (HCONN,             X
*            WOD,                X
*            OPTIONS,           X
*            HOBJ,              X
*            COMPCODE,          X
*            REASON),VL,MF=(E,CALLST)
*
*   LA   R5,MQCC_OK             CHECK THE COMPLETION CODE
*   C    R5,COMPCODE            FROM THE REQUEST AND BRANCH
*   BNE  BADCALL                TO ERROR ROUTINE IF NOT MQCC_OK
*
*   MVC  TEMP_Q,WOD_OBJECTNAME  SAVE NAME OF TEMPORARY Q
*                               CREATED BY OPEN OF MODEL Q
*
*
*
*
*   BADCALL DS    0H
*
*
*
*   CONSTANTS:
*
*   MOD_Q DC    CL48'QUERY.REPLY.MODEL' MODEL QUEUE NAME
*   DYN_Q DC    CL48'QUERY.TEMPQ.*'     DYNAMIC QUEUE NAME
*
*   CMQODA DSECT=NO,LIST=YES  CONSTANT VERSION OF MQOD
*   CMQA                               MQI VALUE EQUATES
*
*   WORKING STORAGE
*
*   DFHEISTG
*
*   HCONN DS F                CONNECTION HANDLE
*   OPTIONS DS F              OPEN OPTIONS
*   HOBJ DS F                 OBJECT HANDLE
*   COMPCODE DS F             MQI COMPLETION CODE
*   REASON DS F               MQI REASON CODE
*   TEMP_Q DS CL(MQ_Q_NAME_LENGTH) SAVED QNAME AFTER OPEN
*
*   WOD    CMQODA DSECT=NO,LIST=YES WORKING VERSION OF MQOD
*
*   CALLST CALL ,(0,0,0,0,0,0,0,0,0,0,0,0),VL,MF=L LIST FORM
*                                               OF CALL
*                                               MACRO
*
*
*
*   END
```

## ***Abrindo uma fila existente***

Este exemplo demonstra como usar a chamada MQOPEN para abrir uma fila já definida.

Ele mostra como especificar duas opções. Essa extração não é obtida dos aplicativos de amostra fornecidos com o IBM MQ.

```

:
*
*   R5 = WORK REGISTER.
*
OPEN   DS   0H
*
*       MVC  WOD_AREA,MQOD_AREA  INITIALIZE WORKING VERSION OF
*                               MQOD WITH DEFAULTS
*       MVC  WOD_OBJECTNAME,Q_NAME  SPECIFY Q NAME TO OPEN
*       LA   R5,MQOO_INPUT_EXCLUSIVE  OPEN FOR MQGET CALLS
*
*       ST   R5,OPTIONS
*
* ISSUE MQI OPEN REQUEST USING REENTRANT FORM
* OF CALL MACRO
*
*       CALL MQOPEN,                X
*           (HCONN,                 X
*            WOD,                   X
*            OPTIONS,               X
*            HOBJ,                 X
*            COMPCODE,             X
*            REASON),VL,MF=(E,CALLST)
*
*       LA   R5,MQCC_OK             CHECK THE COMPLETION CODE
*       C    R5,COMPCODE            FROM THE REQUEST AND BRANCH
*       BNE  BADCALL               TO ERROR ROUTINE IF NOT MQCC_OK
*
*       :
BADCALL DS   0H
:
:
*
*   CONSTANTS:
*
Q_NAME  DC   CL48'REQUEST.QUEUE'  NAME OF QUEUE TO OPEN
*
*       CMQODA DSECT=NO,LIST=YES  CONSTANT VERSION OF MQOD
*       CMQA
*
*   WORKING STORAGE
*
*       DFHEISTG
HCONN   DS F           CONNECTION HANDLE
OPTIONS DS F           OPEN OPTIONS
HOBJ    DS F           OBJECT HANDLE
COMPCODE DS F         MQI COMPLETION CODE
REASON  DS F           MQI REASON CODE
*
WOD     CMQODA DSECT=NO,LIST=YES  WORKING VERSION OF MQOD
*
CALLLST CALL , (0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0),VL,MF=L  LIST FORM
*                                     OF CALL
*                                     MACRO
*
*       :
*       END

```

### ***fechar uma fila***

Este exemplo demonstra como usar a chamada MQCLOSE para fechar uma fila

Essa extração não é obtida dos aplicativos de amostra fornecidos com o IBM MQ.

```

:
*
* ISSUE MQI CLOSE REQUEST USING REENTRANT FROM OF
* CALL MACRO
*
*       HCONN WAS SET BY A PREVIOUS MQCONN REQUEST
*       HOBJ  WAS SET BY A PREVIOUS MQOPEN REQUEST
*       R5 = WORK REGISTER
*
*       CLOSE DS   0H
*       LA   R5,MQCO_NONE          NO SPECIAL CLOSE OPTIONS

```

```

*      ST   R5,OPTIONS          ARE REQUIRED.
*
*      CALL MQCLOSE,           X
*              (HCONN,         X
*              HOBJ,           X
*              OPTIONS,        X
*              COMPCODE,       X
*              REASON),        X
*              VL,MF=(E,CALLST)
*
*      LA   R5,MQCC_OK
*      C    R5,COMPCODE
*      BNE  BADCALL
*
*      :
BADCALL DS   0H
*      :
*              CONSTANTS
*
*      CMQA
*
*      WORKING STORAGE (REentrant)
*
WEG4    DSECT
*
CALLLST CALL , (0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0),VL,MF=L
*
HCONN   DS   F
HOBJ    DS   F
OPTIONS DS   F
COMPCODE DS  F
REASON  DS   F
*
*
LEG4    EQU  *-WKEG4
END

```

### ***Colocando uma mensagem usando MQPUT***

Este exemplo demonstra como usar a chamada MQPUT para colocar uma mensagem em uma fila. Essa extração não é obtida dos aplicativos de amostra fornecidos com o IBM MQ.

```

:
*      CONNECT TO QUEUE MANAGER
*
CONN    DS   0H
:
*
*      OPEN A QUEUE
*
OPEN    DS   0H
:
*
*      R4,R5,R6,R7 = WORK REGISTER.
*
PUT     DS   0H
*      LA   R4,MQMD             SET UP ADDRESSES AND
*      LA   R5,MQMD_LENGTH     LENGTH FOR USE BY MVCL
*      LA   R6,WMD             INSTRUCTION, AS MQMD IS
*      LA   R7,WMD_LENGTH     OVER 256 BYES LONG.
*      MVCL R6,R4             INITIALIZE WORKING VERSION
*                              OF MESSAGE DESCRIPTOR
*
*      MVC  WPMO_AREA,MQPMO_AREA INITIALIZE WORKING MQPMO
*
*
*      LA   R5,BUFFER_LEN     RETRIEVE THE BUFFER LENGTH
*      ST   R5,BUFFLEN       AND SAVE IT FOR MQM USE
*
*      MVC  BUFFER,TEST_MSG   SET THE MESSAGE TO BE PUT
*
*      ISSUE MQI PUT REQUEST USING REENTRANT FORM
*      OF CALL MACRO
*
*      HCONN WAS SET BY PREVIOUS MQCONN REQUEST
*      HOBJ WAS SET BY PREVIOUS MQOPEN REQUEST
*
*      CALL MQPUT,           X

```

```

                (HCONN,                X
                HOBJ,                  X
                WMD,                   X
                WPMO,                  X
                BUFFLEN,               X
                BUFFER,                X
                COMPCODE,              X
                REASON),VL,MF=(E,CALLLST)
*
        LA R5,MQCC_OK
        C  R5,COMPCODE
        BNE BADCALL
*
        :
BADCALL DS 0H
        :

```

```

*
*   CONSTANTS
*
CMQMDA DSECT=NO,LIST=YES,PERSISTENCE=MQPER_PERSISTENT
CMQPMOA DSECT=NO,LIST=YES
CMQA
TEST_MSG DC CL80'THIS IS A TEST MESSAGE'
*
*   WORKING STORAGE DSECT
*
WORKSTG DSECT
*
COMPCODE DS F
REASON   DS F
BUFFLEN  DS F
OPTIONS  DS F
HCONN    DS F
HOBJ     DS F
*
BUFFER   DS CL80
BUFFER_LEN EQU *-BUFFER
*
WMD     CMQMDA DSECT=NO,LIST=NO
WPMO    CMQPMOA DSECT=NO,LIST=NO
*
CALLLST CALL ,(0,0,0,0,0,0,0,0,0,0),VL,MF=L
*
:
END

```

### ***Colocando uma mensagem usando MQPUT1***

Este exemplo demonstra como usar a chamada MQPUT1 para abrir uma fila, colocar uma única mensagem na fila, em seguida, fechar a fila.

Essa extração não é obtida dos aplicativos de amostra fornecidos com o IBM MQ.

```

:
*
*   CONNECT TO QUEUE MANAGER
*
CONN    DS 0H
:
*
*   R4,R5,R6,R7 = WORK REGISTER.
*
PUT     DS 0H
*
        MVC WOD_AREA,MQOD_AREA      INITIALIZE WORKING VERSION OF
*                                     MQOD WITH DEFAULTS
        MVC WOD_OBJECTNAME,Q_NAME   SPECIFY Q NAME FOR PUT1
*
        LA  R4,MQMD                  SET UP ADDRESSES AND
        LA  R5,MQMD_LENGTH           LENGTH FOR USE BY MVCL
        LA  R6,WMD                   INSTRUCTION, AS MQMD IS
        LA  R7,WMD_LENGTH           OVER 256 BYES LONG.
        MVCL R6,R4                  INITIALIZE WORKING VERSION
*                                     OF MESSAGE DESCRIPTOR

```

```

*
*      MVC  WPMO_AREA,MQPMO_AREA      INITIALIZE WORKING MQPMO
*
*      LA   R5,BUFFER_LEN             RETRIEVE THE BUFFER LENGTH
*      ST   R5,BUFFLEN                AND SAVE IT FOR MQM USE
*
*      MVC  BUFFER,TEST_MSG           SET THE MESSAGE TO BE PUT
*
*  * ISSUE MQI PUT REQUEST USING REENTRANT FORM OF CALL MACRO
*
*      HCONN WAS SET BY PREVIOUS MQCONN REQUEST
*      HOBJ WAS SET BY PREVIOUS MQOPEN REQUEST
*
*      CALL MQPUT1,                   X
*          (HCONN,                    X
*           LMQOD,                     X
*           LMQMD,                     X
*           LMQPMO,                    X
*           BUFFERLENGTH,              X
*           BUFFER,                    X
*           COMPCODE,                  X
*           REASON),VL,MF=(E,CALLST)
*
*      LA   R5,MQCC_OK
*      C    R5,COMPCODE
*      BNE BADCALL
*
*      :
BADCALL DS 0H
*
*      :
*

```

```

*      CONSTANTS
*
*      CMQMDA DSECT=NO,LIST=YES,PERSISTENCE=MQPER_PERSISTENT
*      CMQPMOA DSECT=NO,LIST=YES
*      CMQODA DSECT=NO,LIST=YES
*      CMQA
*
*      TEST_MSG DC CL80'THIS IS ANOTHER TEST MESSAGE'
*      Q_NAME   DC CL48'TEST.QUEUE.NAME'
*
*      WORKING STORAGE DSECT
*
*      WORKSTG DSECT
*
*      COMPCODE DS F
*      REASON   DS F
*      BUFFLEN  DS F
*      OPTIONS  DS F
*      HCONN    DS F
*      HOBJ     DS F
*
*      BUFFER   DS CL80
*      BUFFER_LEN EQU *-BUFFER
*
*      WOD      CMQODA DSECT=NO,LIST=YES      WORKING VERSION OF MQOD
*      WMD      CMQMDA DSECT=NO,LIST=NO
*      WPMO     CMQPMOA DSECT=NO,LIST=NO
*
*      CALLST   CALL , (0,0,0,0,0,0,0,0,0,0,0),VL,MF=L
*
*      :
*      END

```

### ***Obtendo uma mensagem***

Este exemplo demonstra como usar a chamada MQGET para remover uma mensagem de uma fila.

Essa extração não é obtida dos aplicativos de amostra fornecidos com o IBM MQ.

```

*
*      CONNECT TO QUEUE MANAGER

```

```

*
CONN    DS   0H
:
*
*      OPEN A QUEUE FOR GET
*
OPEN    DS   0H
:
*
*      R4,R5,R6,R7 = WORK REGISTER.
*
GET     DS   0H
      LA   R4,MQMD                SET UP ADDRESSES AND
      LA   R5,MQMD_LENGTH         LENGTH FOR USE BY MVCL
      LA   R6,WMD                 INSTRUCTION, AS MQMD IS
      LA   R7,WMD_LENGTH         OVER 256 BYES LONG.
      MVCL R6,R4                 INITIALIZE WORKING VERSION
*                                OF MESSAGE DESCRIPTOR
*
*      MVC  WGMO_AREA,MQGMO_AREA  INITIALIZE WORKING MQGMO
*
      LA   R5,BUFFER_LEN         RETRIEVE THE BUFFER LENGTH
      ST   R5,BUFFLEN           AND SAVE IT FOR MQM USE
*
*
*      ISSUE MQI GET REQUEST USING REENTRANT FORM OF CALL MACRO
*
*      HCONN WAS SET BY PREVIOUS MQCONN REQUEST
*      HOBJ WAS SET BY PREVIOUS MQOPEN REQUEST
*
      CALL MQGET,                X
          (HCONN,                X
           HOBJ,                 X
           WMD,                  X
           WGMO,                 X
           BUFFLEN,              X
           BUFFER,               X
           DATALEN,             X
           COMPCODE,             X
           REASON),              X
          VL,MF=(E,CALLLST)
*
      LA   R5,MQCC_OK
      C    R5,COMPCODE
      BNE BADCALL
*
      :
BADCALL DS   0H
:

```

```

*
*      CONSTANTS
*
*      CMQMDA DSECT=NO,LIST=YES
*      CMQGMOA DSECT=NO,LIST=YES
*      CMQA
*
*      WORKING STORAGE DSECT
*
WORKSTG DSECT
*
COMPCODE DS F
REASON   DS F
BUFFLEN  DS F
DATALEN  DS F
OPTIONS  DS F
HCONN    DS F
HOBJ     DS F
*
BUFFER   DS CL80
BUFFER_LEN EQU *-BUFFER
*
WMD      CMQMDA DSECT=NO,LIST=NO
WGMO     CMQGMOA DSECT=NO,LIST=NO
*
CALLLST  CALL , (0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0),VL,MF=L
*

```



```
:\nEND
```

## Obtendo uma mensagem usando a opção de espera

Este exemplo demonstra como usar a opção de espera da chamada MQGET

Este código aceita mensagens truncadas.. Essa extração não é obtida dos aplicativos de amostra fornecidos com o IBM MQ.

```
:\n*\nCONNECT TO QUEUE MANAGER\nCONN DS 0H\n*\nOPEN A QUEUE FOR GET\nOPEN DS 0H\n*\nR4,R5,R6,R7 = WORK REGISTER.\nGET DS 0H\nLA R4,MQMD SET UP ADDRESSES AND\nLA R5,MQMD_LENGTH LENGTH FOR USE BY MVCL\nLA R6,WMD INSTRUCTION, AS MQMD IS\nLA R7,WMD_LENGTH OVER 256 BYES LONG.\nMVCL R6,R4 INITIALIZE WORKING VERSION\n*\nOF MESSAGE DESCRIPTOR\n*\nMVC WGM0_AREA,MQGM0_AREA INITIALIZE WORKING MQGM0\nL R5,=AL4(MQGM0_WAIT)\nA R5,=AL4(MQGM0_ACCEPT_TRUNCATED_MSG)\nST R5,WGM0_OPTIONS\nMVC WGM0_WAITINTERVAL,TWO_MINUTES WAIT UP TO TWO\nMINUTES BEFORE\nFAILING THE\nCALL\n*\nLA R5,BUFFER_LEN RETRIEVE THE BUFFER LENGTH\nST R5,BUFFLEN AND SAVE IT FOR MQM USE\n*\n*\nISSUE MQI GET REQUEST USING REENTRANT FORM OF CALL MACRO\n*\n*\nHCONN WAS SET BY PREVIOUS MQCONN REQUEST\nHOBJ WAS SET BY PREVIOUS MQOPEN REQUEST\n*\nCALL MQGET, X\n(HCONN, X\nHOBJ, X\nWMD, X\nWGM0, X\nBUFFLEN, X\nBUFFER, X\nDATALEN, X\nCOMPCODE, X\nREASON), X\nVL,MF=(E,CALLLST)\n*\nLA R5,MQCC_OK DID THE MQGET REQUEST\nC R5,COMPCODE WORK OK?\nBE GETOK YES, SO GO AND PROCESS.\nLA R5,MQCC_WARNING NO, SO CHECK FOR A WARNING.\nC R5,COMPCODE IS THIS A WARNING?\nBE CHECK_W YES, SO CHECK THE REASON.\n*\nLA R5,MQRC_NO_MSG_AVAILABLE IT MUST BE AN ERROR.\nIS IT DUE TO AN EMPTY\nC R5,REASON QUEUE?\nBE NOMSG YES, SO HANDLE THE ERROR\nB BADCALL NO, SO GO TO ERROR ROUTINE\n*\nCHECK_W DS 0H\nLA R5,MQRC_TRUNCATED_MSG_ACCEPTED IS THIS A\nTRUNCATED\nC R5,REASON MESSAGE?\nBE GETOK YES, SO GO AND PROCESS.\nB BADCALL NO, SOME OTHER WARNING\n*\nNOMSG DS 0H
```

```

:
GETOK DS 0H
:

```

```

BADCALL DS 0H
:
*
*   CONSTANTS
*
*   CMQMDA DSECT=NO,LIST=YES
*   CMQMOA DSECT=NO,LIST=YES
*   CMQA
*
* TWO_MINUTES DC F'120000'      GET WAIT INTERVAL
*
*   WORKING STORAGE DSECT

```

```

*
WORKSTG DSECT
*
COMPCODE DS F
REASON DS F
BUFFLEN DS F
DATALEN DS F
OPTIONS DS F
HCONN DS F
HOBJ DS F
*
BUFFER DS CL80
BUFFER_LEN EQU *-BUFFER
*
WMD CMQMDA DSECT=NO,LIST=NO
WMO CMQMOA DSECT=NO,LIST=NO
*
CALLLST CALL ,(0,0,0,0,0,0,0,0,0,0),VL,MF=L
*
:
END

```

### ***Obtendo uma mensagem usando sinalização***

Este exemplo demonstra como usar a chamada MQGET para configurar um sinal para que você seja notificado quando uma mensagem adequada chegar em uma fila

Essa extração não é obtida dos aplicativos de amostra fornecidos com o IBM MQ.

```

:
*
*   CONNECT TO QUEUE MANAGER
*
CONN DS 0H
:
*
*   OPEN A QUEUE FOR GET
*
OPEN DS 0H
:
*
*   R4,R5,R6,R7 = WORK REGISTER.
*
GET DS 0H
LA R4,MQMD          SET UP ADDRESSES AND
LA R5,MQMD_LENGTH  LENGTH FOR USE BY MVCL
LA R6,WMD           INSTRUCTION, AS MQMD IS
LA R7,WMD_LENGTH   OVER 256 BYES LONG.
MVCL R6,R4         INITIALIZE WORKING VERSION
*                   OF MESSAGE DESCRIPTOR

```

```

*
MVC WMO_AREA,MQMO_AREA  INITIALIZE WORKING MQMO
LA R5,MQMO_SET_SIGNAL
ST R5,WMO_OPTIONS
MVC WMO_WAITINTERVAL,FIVE_MINUTES  WAIT UP TO FIVE

```

```

*                                     MINUTES BEFORE
*                                     FAILING THE CALL
*
XC  SIG_ECB,SIG_ECB  CLEAR THE ECB
LA  R5,SIG_ECB      GET THE ADDRESS OF THE ECB
ST  R5,WGMO_SIGNAL1 AND PUT IT IN THE WORKING
*                                     MQGMO
*
LA  R5,BUFFER_LEN   RETRIEVE THE BUFFER LENGTH
ST  R5,BUFFLEN      AND SAVE IT FOR MQM USE
*
*
*   ISSUE MQI GET REQUEST USING REENTRANT FORM OF CALL MACRO
*
*   HCONN WAS SET BY PREVIOUS MQCONN REQUEST
*   HOBJ WAS SET BY PREVIOUS MQOPEN REQUEST
*
CALL  MQGET,                X
      (HCONN,                X
       HOBJ,                  X
       WMD,                   X
       WGMO,                   X
       BUFFLEN,               X
       BUFFER,                 X
       DATALEN,              X
       COMPCODE,              X
       REASON),              X
      VL,MF=(E,CALLLST)
*
LA  R5,MQCC_OK          DID THE MQGET REQUEST
C   R5,COMPCODE         WORK OK?
BE  GETOK              YES, SO GO AND PROCESS.
LA  R5,MQCC_WARNING    NO, SO CHECK FOR A WARNING.
C   R5,COMPCODE         IS THIS A WARNING?
BE  CHECK_W           YES, SO CHECK THE REASON.
B   BADCALL            NO, SO GO TO ERROR ROUTINE
*

```

```

CHECK_W DS 0H
LA R5,MQRC_SIGNAL_REQUEST_ACCEPTED
C R5,REASON SIGNAL REQUEST SIGNAL SET?
BNE BADCALL NO, SOME ERROR OCCURRED
B DOWORK YES, SO DO SOMETHING
*                                     ELSE
*
CHECKSIG DS 0H
CLC SIG_ECB+1(3),=AL3(MQEC_MSG_ARRIVED)
*                                     IS A MESSAGE AVAILABLE?
BE GET YES, SO GO AND GET IT
*
CLC SIG_ECB+1(3),=AL3(MQEC_WAIT_INTERVAL_EXPIRED)
*                                     HAVE WE WAITED LONG ENOUGH?
BE NOMSG YES, SO SAY NO MSG AVAILABLE
B BADCALL IF IT'S ANYTHING ELSE
*                                     GO TO ERROR ROUTINE.
*
DOWORK DS 0H
*
*   TM SIG_ECB,X'40' HAS THE SIGNAL ECB BEEN POSTED?
BO CHECKSIG YES, SO GO AND CHECK WHY
B DOWORK NO, SO GO AND DO MORE WORK
*
NOMSG DS 0H
*
GETOK DS 0H
*
BADCALL DS 0H
*
*
*   CONSTANTS
*
CMQMDA DSECT=NO,LIST=YES
CMQMOA DSECT=NO,LIST=YES
CMQA
*
FIVE_MINUTES DC F'300000' GET SIGNAL INTERVAL
*
*   WORKING STORAGE DSECT
*

```

```

WORKSTG DSECT
*
COMPCODE DS F
REASON DS F
BUFFLEN DS F
DATALEN DS F
OPTIONS DS F
HCONN DS F
HOBJ DS F
SIG_ECB DS F

```

```

*
BUFFER DS CL80
BUFFER_LEN EQU *-BUFFER
*
WMD CMQMDA DSECT=NO,LIST=NO
WGMO CMQGMOA DSECT=NO,LIST=NO
*
CALLLIST CALL ,(0,0,0,0,0,0,0,0,0,0,0,0),VL,MF=L
*
:
END

```

### Consultando e configurando os atributos de uma fila

Este exemplo demonstra como usar a chamada MQINQ para consultar os atributos de uma fila e usar a chamada MQSET para alterar os atributos de uma fila.

Essa extração é obtida do aplicativo de amostra Atributos da Fila (programa CSQ4CAC1) fornecido com IBM MQ for z/OS.

```

:
DFHEISTG DSECT
:
OBJDESC CMQODA LIST=YES Working object descriptor
*
SELECTORCOUNT DS F Number of selectors
INTATTRCOUNT DS F Number of integer attributes
CHARATTRLENGTH DS F char attributes length
CHARATTRS DS C Area for char attributes
*
OPTIONS DS F Command options
HCONN DS F Handle of connection
HOBJ DS F Handle of object
COMPCODE DS F Completion code
REASON DS F Reason code
SELECTOR DS 2F Array of selectors
INTATTRS DS 2F Array of integer attributes
:
OBJECT DS CL(MQ_Q_NAME_LENGTH) Name of queue
:
CALLLIST CALL ,(0,0,0,0,0,0,0,0,0,0,0,0),VL,MF=L
*****
* PROGRAM EXECUTION STARTS HERE *
:
CSQ4CAC1 DFHEIENT CODEREG=(R3),DATAREG=(R13)
:
* Initialize the variables for the set call
*
SR R0,R0 Clear register zero
ST R0,CHARATTRLENGTH Set char length to zero
LA R0,2 Load to set
ST R0,SELECTORCOUNT selectors add
ST R0,INTATTRCOUNT integer attributes
*
LA R0,MQIA_INHIBIT_GET Load q attribute selector
ST R0,SELECTOR+0 Place in field
LA R0,MQIA_INHIBIT_PUT Load q attribute selector
ST R0,SELECTOR+4 Place in field
*
UPDTEST DS 0H
CLC ACTION,CINHIB Are we inhibiting?
BE UPDINHBT Yes branch to section
*
CLC ACTION,CALLOW Are we allowing?
BE UPDALLOW Yes branch to section

```

```

*          MVC M00_MSG,M01_MSG1   Invalid request
          BR  R6                   Return to caller
*

```

```

UPDINHBT DS 0H
          MVC UPDTYPE,CINHIBIT     Indicate action type
          LA  R0,MQQA_GET_INHIBITED Load attribute value
          ST  R0,INTATTRS+0        Place in field
          LA  R0,MQQA_PUT_INHIBITED Load attribute value
          ST  R0,INTATTRS+4        Place in field
          B   UPDCALL              Go and do call

```

```

*
UPDALLOW DS 0H
          MVC UPDTYPE,CALLOWED     Indicate action type
          LA  R0,MQQA_GET_ALLOWED   Load attribute value
          ST  R0,INTATTRS+0        Place in field
          LA  R0,MQQA_PUT_ALLOWED   Load attribute value
          ST  R0,INTATTRS+4        Place in field
          B   UPDCALL              Go and do call

```

```

*
UPDCALL  DS 0H
          CALL MQSET,              C
              (HCONN,             C
              HOBJ,               C
              SELECTORCOUNT,    C
              SELECTOR,          C
              INTATTRCOUNT,     C
              INTATTRS,          C
              CHARATTRLENGTH,    C
              CHARATTRS,         C
              COMPCODE,          C
              REASON),           C
          VL,MF=(E,CALLLIST)

```

```

*          LA  R0,MQCC_OK          Load expected compcode
          C   R0,COMPCODE          Was set successful?
          :

```

```

* SECTION NAME : INQUIRE          *
* FUNCTION     : Inquires on the objects attributes *
* CALLED BY    : PROCESS           *
* CALLS        : OPEN, CLOSE, CODES *
* RETURN       : To Register 6     *

```

```

INQUIRE DS 0H
          :

```

```

*          Initialize the variables for the inquire call

```

```

*          SR  R0,R0               Clear register zero
          ST  R0,CHARATTRLENGTH    Set char length to zero
          LA  R0,2                 Load to set
          ST  R0,SELECTORCOUNT    selectors add
          ST  R0,INTATTRCOUNT     integer attributes

```

```

*          LA  R0,MQIA_INHIBIT_GET Load attribute value
          ST  R0,SELECTOR+0        Place in field
          LA  R0,MQIA_INHIBIT_PUT Load attribute value
          ST  R0,SELECTOR+4        Place in field
          CALL MQINQ,              C
              (HCONN,             C
              HOBJ,               C
              SELECTORCOUNT,    C
              SELECTOR,          C
              INTATTRCOUNT,     C
              INTATTRS,          C
              CHARATTRLENGTH,    C
              CHARATTRS,         C
              COMPCODE,          C
              REASON),           C
          VL,MF=(E,CALLLIST)
          LA  R0,MQCC_OK          Load expected compcode
          C   R0,COMPCODE          Was inquire successful?
          :

```

## exemplos de PL/I

O uso de PL/I é suportado apenas pelo z/OS Esta coleção de tópicos demonstra técnicas usando exemplos de PL/I.

### **Conectando-se a um gerenciador de filas**

Este exemplo demonstra como usar a chamadas MQCONN para conectar um programa a um gerenciador de filas em lote z/OS

Essa extração não é obtida dos aplicativos de amostra fornecidos com o IBM MQ.

```
%INCLUDE SYSLIB(CMQP);
%INCLUDE SYSLIB(CMQEPP);
:
/*****
/* STRUCTURE BASED ON PARAMETER INPUT AREA (PARAM) */
*****/
DCL 1 INPUT_PARAM      BASED(ADDR(PARAM)),
    2 PARAM_LENGTH     FIXED BIN(15),
    2 PARAM_MQMNNAME   CHAR(48);
:

/*****
/* WORKING STORAGE DECLARATIONS */
*****/
DCL MQMNNAME           CHAR(48);
DCL COMPCODE           BINARY FIXED (31);
DCL REASON             BINARY FIXED (31);
DCL HCONN             BINARY FIXED (31);
:

/*****
/* COPY QUEUE MANAGER NAME PARAMETER */
/* TO LOCAL STORAGE */
*****/
MQMNNAME = ' ';
MQMNNAME = SUBSTR(PARAM_MQMNNAME,1,PARAM_LENGTH);
:

/*****
/* CONNECT FROM THE QUEUE MANAGER */
*****/
CALL MQCONN (MQMNNAME, /* MQM SYSTEM NAME */
            HCONN, /* CONNECTION HANDLE */
            COMPCODE, /* COMPLETION CODE */
            REASON); /* REASON CODE */

/*****
/* TEST THE COMPLETION CODE OF THE CONNECT CALL. */
/* IF THE CALL HAS FAILED ISSUE AN ERROR MESSAGE */
/* SHOWING THE COMPLETION CODE AND THE REASON CODE. */
*****/
IF COMPCODE = MQCC_OK
  THEN DO;
  :
  CALL ERROR_ROUTINE;
END;
```

### **desconectando-se de um gerenciador de filas**

Este exemplo demonstra como usar a chamada MQDISC para desconectar um programa de um gerenciador de fila em lote do z/OS

Essa extração não é obtida dos aplicativos de amostra fornecidos com o IBM MQ.

```
%INCLUDE SYSLIB(CMQP);
%INCLUDE SYSLIB(CMQEPP);
:
/*****
/* WORKING STORAGE DECLARATIONS */
*****/
DCL COMPCODE           BINARY FIXED (31);
DCL REASON             BINARY FIXED (31);
DCL HCONN             BINARY FIXED (31);
:

/*****
/* DISCONNECT FROM THE QUEUE MANAGER */
*****/
```

```

/*****/
CALL MQDISC (HCONN,      /* CONNECTION HANDLE */
             COMPCODE,   /* COMPLETION CODE   */
             REASON);    /* REASON CODE       */
.
/*****/
/* TEST THE COMPLETION CODE OF THE DISCONNECT CALL. */
/* IF THE CALL HAS FAILED ISSUE AN ERROR MESSAGE  */
/* SHOWING THE COMPLETION CODE AND THE REASON CODE.*/
/*****/
IF COMPCODE = MQCC_OK
  THEN DO;
  .
  CALL ERROR_ROUTINE;
END;

```

### ***Criando uma fila dinâmica***

Este exemplo demonstra como usar a chamada MQOPEN para criar uma fila dinâmica..

Essa extração não é obtida dos aplicativos de amostra fornecidos com o IBM MQ.

```

%INCLUDE SYSLIB(CMQP);
%INCLUDE SYSLIB(CMQEPP);
.
/*****/
/* WORKING STORAGE DECLARATIONS */
/*****/
DCL COMPCODE          BINARY FIXED (31);
DCL REASON            BINARY FIXED (31);
DCL HCONN             BINARY FIXED (31);
DCL HOBJ              BINARY FIXED (31);
DCL OPTIONS           BINARY FIXED (31);
.
DCL MODEL_QUEUE_NAME CHAR(48) INIT('PL1.REPLY.MODEL');
DCL DYNAMIC_NAME_PREFIX CHAR(48) INIT('PL1.TEMPQ.*');
DCL DYNAMIC_QUEUE_NAME CHAR(48) INIT(' ');
.
/*****/
/* LOCAL COPY OF OBJECT DESCRIPTOR */
/*****/
DCL 1 LMQOD LIKE MQOD;
.
/*****/
/* SET UP OBJECT DESCRIPTOR FOR OPEN OF REPLY QUEUE */
/*****/
LMQOD.OBJECTTYPE =MQOT_Q;
LMQOD.OBJECTNAME = MODEL_QUEUE_NAME;
LMQOD.DYNAMICQNAME = DYNAMIC_NAME_PREFIX;
OPTIONS = MQOO_INPUT_EXCLUSIVE;

CALL MQOPEN (HCONN,
             LMQOD,
             OPTIONS,
             HOBJ,
             COMPCODE,
             REASON);

/*****/
/* TEST THE COMPLETION CODE OF THE OPEN CALL. */
/* IF THE CALL HAS FAILED ISSUE AN ERROR MESSAGE */
/* SHOWING THE COMPLETION CODE AND THE REASON CODE. */
/* IF THE CALL HAS SUCCEEDED THEN EXTRACT THE NAME OF */
/* THE NEWLY CREATED DYNAMIC QUEUE FROM THE OBJECT */
/* DESCRIPTOR. */
/*****/
IF COMPCODE = MQCC_OK
  THEN DO;
  .
  CALL ERROR_ROUTINE;
END;
ELSE
  DYNAMIC_QUEUE_NAME = LMQOD_OBJECTNAME;

```

### ***Abrindo uma fila existente***

Este exemplo demonstra como usar a chamada MQOPEN para abrir uma fila existente..

Essa extração não é obtida dos aplicativos de amostra fornecidos com o IBM MQ.

```
%INCLUDE SYSLIB(CMQP);
%INCLUDE SYSLIB(CMQEPP);
:
/*****
/* WORKING STORAGE DECLARATIONS */
*****/
DCL COMPCODE          BINARY FIXED (31);
DCL REASON            BINARY FIXED (31);
DCL HCONN             BINARY FIXED (31);
DCL HOBJ              BINARY FIXED (31);
DCL OPTIONS           BINARY FIXED (31);
:
DCL QUEUE_NAME        CHAR(48) INIT('PL1.LOCAL.QUEUE');
:
/*****
/* LOCAL COPY OF OBJECT DESCRIPTOR */
*****/
DCL 1 LMQOD LIKE MQOD;
:
/*****
/* SET UP OBJECT DESCRIPTOR FOR OPEN OF REPLY QUEUE */
*****/
LMQOD.OBJECTTYPE = MQOT_Q;
LMQOD.OBJECTNAME = QUEUE_NAME;
OPTIONS = MQOO_INPUT_EXCLUSIVE;

CALL MQOPEN (HCONN,
             LMQOD,
             OPTIONS,
             HOBJ,
             COMPCODE,
             REASON);

/*****
/* TEST THE COMPLETION CODE OF THE OPEN CALL. */
/* IF THE CALL HAS FAILED ISSUE AN ERROR MESSAGE */
/* SHOWING THE COMPLETION CODE AND THE REASON CODE. */
*****/
IF COMPCODE /= MQCC_OK
  THEN DO;
  :
  CALL ERROR_ROUTINE;
END;
```

## ***fechar uma fila***

Este exemplo demonstra como usar a chamada MQCLOSE

Essa extração não é obtida dos aplicativos de amostra fornecidos com o IBM MQ.

```
%INCLUDE SYSLIB(CMQP);
%INCLUDE SYSLIB(CMQEPP);
:
/*****
/* WORKING STORAGE DECLARATIONS */
*****/
DCL COMPCODE          BINARY FIXED (31);
DCL REASON            BINARY FIXED (31);
DCL HCONN             BINARY FIXED (31);
DCL HOBJ              BINARY FIXED (31);
DCL OPTIONS           BINARY FIXED (31);
:
/*****
/* SET CLOSE OPTIONS */
*****/
OPTIONS=MQCO_NONE;

/*****
/* CLOSE QUEUE */
*****/
CALL MQCLOSE (HCONN, /* CONNECTION HANDLE */
             HOBJ, /* OBJECT HANDLE */
             OPTIONS, /* CLOSE OPTIONS */
             COMPCODE, /* COMPLETION CODE */
             REASON); /* REASON CODE */
```



```

/*****
/* TEST THE COMPLETION CODE OF THE CLOSE CALL.          */
/* IF THE CALL HAS FAILED ISSUE AN ERROR MESSAGE        */
/* SHOWING THE COMPLETION CODE AND THE REASON CODE.     */
/*****
    IF COMPCODE = MQCC_OK
        THEN DO;
            :
            CALL ERROR_ROUTINE;
        END;

```

## Colocando uma mensagem usando MQPUT

Este exemplo demonstra como usar a chamada MQPUT usando o contexto

Essa extração não é obtida dos aplicativos de amostra fornecidos com o IBM MQ.

```

%INCLUDE SYSLIB(CMQP);
%INCLUDE SYSLIB(CMQEPP);
:
/*****
/* WORKING STORAGE DECLARATIONS                          */
/*****
DCL COMPCODE          BINARY FIXED (31);
DCL REASON            BINARY FIXED (31);
DCL HCONN             BINARY FIXED (31);
DCL HOBJ              BINARY FIXED (31);
DCL OPTIONS           BINARY FIXED (31);
DCL BUFFLEN           BINARY FIXED (31);
DCL BUFFER            CHAR(80);
:
DCL PL1_TEST_MESSAGE CHAR(80)
INIT('***** THIS IS A TEST MESSAGE *****');
:
/*****
/* LOCAL COPY OF MESSAGE DESCRIPTOR                      */
/* AND PUT MESSAGE OPTIONS                              */
/*****
DCL 1 LMQMD LIKE MQMD;
DCL 1 LMQPMO LIKE MQPMO;
:
/*****
/* SET UP MESSAGE DESCRIPTOR                            */
/*****
LMQMD.MSGTYPE = MQMT_DATAGRAM;
LMQMD.PRIORITY = 1;
LMQMD.PERSISTENCE = MQPER_PERSISTENT;
LMQMD.REPLYTOQ = ' ';
LMQMD.REPLYTOQMGR = ' ';
LMQMD.MSGID = MQMI_NONE;
LMQMD.CORRELID = MQCI_NONE;

/*****
/* SET UP PUT MESSAGE OPTIONS                            */
/*****
LMQPMO.OPTIONS = MQPMO_NO_SYNCPOINT;

/*****
/* SET UP LENGTH OF MESSAGE BUFFER AND THE MESSAGE      */
/*****
BUFFLEN = LENGTH(BUFFER);
BUFFER = PL1_TEST_MESSAGE;
/*****
/*
/* HCONN WAS SET BY PREVIOUS MQCONN REQUEST.            */
/* HOBJ WAS SET BY PREVIOUS MQOPEN REQUEST.             */
/*
/*****
CALL MQPUT (HCONN,
            HOBJ,
            LMQMD,
            LMQPMO,
            BUFFLEN,
            BUFFER,
            COMPCODE,
            REASON);

```

```

/*****/
/* TEST THE COMPLETION CODE OF THE PUT CALL. */
/* IF THE CALL HAS FAILED ISSUE AN ERROR MESSAGE */
/* SHOWING THE COMPLETION CODE AND THE REASON CODE. */
/*****/
    IF COMPCODE = MQCC_OK
        THEN DO;
            :
            :
            CALL ERROR_ROUTINE;
        END;

```

## Colocando uma mensagem usando MQPUT1

Este exemplo demonstra como usar a chamada MQPUT1

Essa extração não é obtida dos aplicativos de amostra fornecidos com o IBM MQ.

```

%INCLUDE SYSLIB(CMQEPP);
%INCLUDE SYSLIB(CMQP);
:
/*****/
/* WORKING STORAGE DECLARATIONS */
/*****/
DCL COMPCODE      BINARY FIXED (31);
DCL REASON        BINARY FIXED (31);
DCL HCONN         BINARY FIXED (31);
DCL OPTIONS       BINARY FIXED (31);
DCL BUFFLEN      BINARY FIXED (31);
DCL BUFFER        CHAR(80);
:
DCL REPLY_TO_QUEUE CHAR(48) INIT('PL1.REPLY.QUEUE');
DCL QUEUE_NAME     CHAR(48) INIT('PL1.LOCAL.QUEUE');
DCL PL1_TEST_MESSAGE CHAR(80)
INIT('***** THIS IS ANOTHER TEST MESSAGE *****');
:
/*****/
/* LOCAL COPY OF OBJECT DESCRIPTOR, MESSAGE DESCRIPTOR */
/* AND PUT MESSAGE OPTIONS */
/*****/
DCL 1 LMQOD LIKE MQOD;
DCL 1 LMQMD LIKE MQMD;
DCL 1 LMQPMO LIKE MQPMO;
:
/*****/
/* SET UP OBJECT DESCRIPTOR AS REQUIRED. */
/*****/
LMQOD.OBJECTTYPE = MQOT_Q;
LMQOD.OBJECTNAME = QUEUE_NAME;

/*****/
/* SET UP MESSAGE DESCRIPTOR AS REQUIRED. */
/*****/
LMQMD.MSGTYPE = MQMT_REQUEST;
LMQMD.PRIORITY = 5;
LMQMD.PERSISTENCE = MQPER_PERSISTENT;
LMQMD.REPLYTOQ = REPLY_TO_QUEUE;
LMQMD.REPLYTOQMGR = 'T';
LMQMD.MSGID = MQMI_NONE;
LMQMD.CORRELID = MQCI_NONE;

/*****/
/* SET UP PUT MESSAGE OPTIONS AS REQUIRED */
/*****/
LMQPMO.OPTIONS = MQPMO_NO_SYNCPOINT;

/*****/
/* SET UP LENGTH OF MESSAGE BUFFER AND THE MESSAGE */
/*****/
BUFFLEN = LENGTH(BUFFER);
BUFFER = PL1_TEST_MESSAGE;

CALL MQPUT1 (HCONN,
             LMQOD,
             LMQMD,
             LMQPMO,

```

```

BUFFLEN,
BUFFER,
COMPCODE,
REASON);

/*****/
/* TEST THE COMPLETION CODE OF THE PUT1 CALL. */
/* IF THE CALL HAS FAILED ISSUE AN ERROR MESSAGE SHOWING */
/* THE COMPLETION CODE AND THE REASON CODE. */
/*****/
IF COMPCODE = MQCC_OK
THEN DO;
:
CALL ERROR_ROUTINE;
END;

```

## Obtendo uma mensagem

Este exemplo demonstra como usar a chamada MQGET para remover uma mensagem de uma fila.

Essa extração não é obtida dos aplicativos de amostra fornecidos com o IBM MQ.

```

%INCLUDE SYSLIB(CMQP);
%INCLUDE SYSLIB(CMQEPP);
:
/*****/
/* WORKING STORAGE DECLARATIONS */
/*****/
DCL COMPCODE          BINARY FIXED (31);
DCL REASON            BINARY FIXED (31);
DCL HCONN            BINARY FIXED (31);
DCL HOBJ             BINARY FIXED (31);
DCL BUFFLEN          BINARY FIXED (31);
DCL DATALEN         BINARY FIXED (31);
DCL BUFFER           CHAR(80);
:

/*****/
/* LOCAL COPY OF MESSAGE DESCRIPTOR AND */
/* GET MESSAGE OPTIONS */
/*****/
DCL 1 LMQMD LIKE MQMD;
DCL 1 LMQGMO LIKE MQGMO;
:

/*****/
/* SET UP MESSAGE DESCRIPTOR AS REQUIRED. */
/* MSGID AND CORRELID IN MQMD SET TO NULLS SO FIRST */
/* AVAILABLE MESSAGE WILL BE RETRIEVED. */
/*****/
LMQMD.MSGID = MQMI_NONE;
LMQMD.CORRELID = MQCI_NONE;

/*****/
/* SET UP GET MESSAGE OPTIONS AS REQUIRED. */
/*****/
LMQGMO.OPTIONS = MQGMO_NO_SYNCPOINT;

/*****/
/* SET UP LENGTH OF MESSAGE BUFFER. */
/*****/
BUFFLEN = LENGTH(BUFFER);

/*****/
/*
/* HCONN WAS SET BY PREVIOUS MQCONN REQUEST.
/* HOBJ WAS SET BY PREVIOUS MQOPEN REQUEST.
/*
/*****/

CALL MQGET (HCONN,
           HOBJ,
           LMQMD,
           LMQGMO,
           BUFFERLEN,
           BUFFER,
           DATALEN,
           COMPCODE,
           REASON);

```

```

/*****/
/* TEST THE COMPLETION CODE OF THE GET CALL. */
/* IF THE CALL HAS FAILED ISSUE AN ERROR MESSAGE */
/* SHOWING THE COMPLETION CODE AND THE REASON CODE. */
/*****/
IF COMPCODE = MQCC_OK
  THEN DO;
  :
  CALL ERROR_ROUTINE;
END;

```

## Obtendo uma mensagem usando a opção de espera

Este exemplo demonstra como usar a chamada MQGET com a opção de espera e aceitar mensagens truncadas.

Essa extração não é obtida dos aplicativos de amostra fornecidos com o IBM MQ.

```

%INCLUDE SYSLIB(CMQP);
%INCLUDE SYSLIB(CMQEPP);
:
/*****/
/* WORKING STORAGE DECLARATIONS */
/*****/
DCL COMPCODE          BINARY FIXED (31);
DCL REASON            BINARY FIXED (31);
DCL HCONN             BINARY FIXED (31);
DCL HOBJ              BINARY FIXED (31);
DCL BUFFLEN          BINARY FIXED (31);
DCL DATALEN         BINARY FIXED (31);
DCL BUFFER            CHAR(80);
:
/*****/
/* LOCAL COPY OF MESSAGE DESCRIPTOR AND GET MESSAGE */
/* OPTIONS */
/*****/
DCL 1 LMQMD LIKE MQMD;
DCL 1 LMQGMO LIKE MQGMO;
:
/*****/
/* SET UP MESSAGE DESCRIPTOR AS REQUIRED. */
/* MSGID AND CORRELID IN MQMD SET TO NULLS SO FIRST */
/* AVAILABLE MESSAGE WILL BE RETRIEVED. */
/*****/
LMQMD.MSGID = MQMI_NONE;
LMQMD.CORRELID = MQCI_NONE;

/*****/
/* SET UP GET MESSAGE OPTIONS AS REQUIRED. */
/* WAIT INTERVAL SET TO ONE MINUTE. */
/*****/
LMQGMO.OPTIONS = MQGMO_WAIT +
                MQGMO_ACCEPT_TRUNCATED_MSG +
                MQGMO_NO_SYNCPOINT;
LMQGMO.WAITINTERVAL=60000;

/*****/
/* SET UP LENGTH OF MESSAGE BUFFER. */
/*****/
BUFFLEN = LENGTH(BUFFER);

/*****/
/*
/* HCONN WAS SET BY PREVIOUS MQCONN REQUEST. */
/* HOBJ WAS SET BY PREVIOUS MQOPEN REQUEST. */
/*
/*****/

CALL MQGET (HCONN,
            HOBJ,
            LMQMD,
            LMQGMO,
            BUFFERLEN,
            BUFFER,
            DATALEN,
            COMPCODE,

```

```

REASON);

/*****
/* TEST THE COMPLETION CODE OF THE GET CALL.          */
/* TAKE APPROPRIATE ACTION BASED ON COMPLETION CODE AND */
/* REASON CODE.                                       */
*****/

SELECT (COMPCODE);
  WHEN (MQCC_OK) DO; /* GET WAS SUCCESSFUL */
  :
  END;
  WHEN (MQCC_WARNING) DO;
  IF REASON = MQRC_TRUNCATED_MSG_ACCEPTED
  THEN DO; /* GET WAS SUCCESSFUL */
  :
  END;
  ELSE DO;
  :
  CALL ERROR_ROUTINE;
  END;
  END;
  WHEN (MQCC_FAILED) DO;
  :
  CALL ERROR_ROUTINE;
  END;
  END;
  OTHERWISE;
END;

```

### ***Obtendo uma mensagem usando sinalização***

Uma extração de código que demonstra como usar a chamada MQGET com sinalização..

### **A sinalização está disponível apenas com o IBM MQ for z/OS**

Essa extração não é obtida dos aplicativos de amostra fornecidos com o IBM MQ.

```

%INCLUDE SYSLIB(CMQP);
%INCLUDE SYSLIB(CMQEPP);
:
/*****
/* WORKING STORAGE DECLARATIONS          */
*****/
DCL COMPCODE          BINARY FIXED (31);
DCL REASON            BINARY FIXED (31);
DCL HCONN             BINARY FIXED (31);
DCL HOBJ              BINARY FIXED (31);
DCL DATALEN         BINARY FIXED (31);
DCL BUFFLEN          BINARY FIXED (31);
DCL BUFFER           CHAR(80);
:
DCL ECB_FIXED        FIXED BIN(31);
DCL 1 ECB_OVERLAY   BASED(ADDR(ECB_FIXED)),
      3 ECB_WAIT    BIT,
      3 ECB_POSTED  BIT,
      3 ECB_FLAG3_8 BIT(6),
      3 ECB_CODE    PIC'999';
:
/*****
/* LOCAL COPY OF MESSAGE DESCRIPTOR AND GET MESSAGE */
/* OPTIONS                                          */
*****/
DCL 1 LMQMD LIKE MQMD;
DCL 1 LMQGMO LIKE MQGMO;
:
/*****
/* CLEAR ECB FIELD.                               */
*****/
ECB_FIXED = 0;
:
/*****
/* SET UP MESSAGE DESCRIPTOR AS REQUIRED.          */
/* MSGID AND CORRELID IN MQMD SET TO NULLS SO FIRST */
/* AVAILABLE MESSAGE WILL BE RETRIEVED.          */
*****/
LMQMD.MSGID = MQMI_NONE;
LMQMD.CORRELID = MQCI_NONE;
/*****

```

```

/* SET UP GET MESSAGE OPTIONS AS REQUIRED.          */
/* WAIT INTERVAL SET TO ONE MINUTE.                */
/*****
  LMQGMO.OPTIONS = MQGMO_SET_SIGNAL +
                  MQGMO_NO_SYNCPOINT;
  LMQGMO.WAITINTERVAL=60000;
  LMQGMO.SIGNAL1 = ADDR(ECB_FIXED);
*****/

```

```

/*****
/* SET UP LENGTH OF MESSAGE BUFFER.                */
/* CALL MESSAGE RETRIEVAL ROUTINE.                 */
/*****
  BUFFLEN = LENGTH(BUFFER);
  CALL GET_MSG;
*****/

```

```

/*****
/* TEST THE COMPLETION CODE OF THE GET CALL.       */
/* TAKE APPROPRIATE ACTION BASED ON COMPLETION CODE AND REASON CODE.
/*****

```

```

SELECT;
  WHEN ((COMPCODE = MQCC_OK) &
        (REASON = MQCC_NONE)) DO
    :
    CALL MSG_ROUTINE;
    :
  END;
  WHEN ((COMPCODE = MQCC_WARNING) &
        (REASON = MQRC_SIGNAL_REQUEST_ACCEPTED)) DO;
    :
    CALL DO_WORK;
    :
  END;
  WHEN ((COMPCODE = MQCC_FAILED) &
        (REASON = MQRC_SIGNAL_OUTSTANDING)) DO;
    :
    CALL DO_WORK;
    :
  END;
  OTHERWISE DO;          /* FAILURE CASE */
/*****
/* ISSUE AN ERROR MESSAGE SHOWING THE COMPLETION CODE
/* AND THE REASON CODE.
/*****
  :
  CALL ERROR_ROUTINE;
  :
  END;
END;
:

```

```

DO_WORK: PROC;
:
  IF ECB_POSTED
  THEN DO;
    SELECT(ECB_CODE);
    WHEN(MQEC_MSG_ARRIVED) DO;
      :
      CALL GET_MSG;
      :
    END;
    WHEN(MQEC_WAIT_INTERVAL_EXPIRED) DO;
      :
      CALL NO_MSG;
      :
    END;
    OTHERWISE DO;          /* FAILURE CASE */
/*****
/* ISSUE AN ERROR MESSAGE SHOWING THE COMPLETION CODE
/* AND THE REASON CODE.
/*****
  :
  CALL ERROR_ROUTINE;
  :
  END;

```

```

        END;
    END;
    :
END DO_WORK;
GET_MSG: PROC;

```

```

/*****/
/*
/* HCONN WAS SET BY PREVIOUS MQCONN REQUEST.          */
/* HOBJ WAS SET BY PREVIOUS MQOPEN REQUEST.          */
/* MD AND GMO SET UP AS REQUIRED.                    */
/*
/*****/

        CALL MQGET (HCONN,
                    HOBJ,
                    LMQMD,
                    LMQGMO,
                    BUFLLEN,
                    BUFFER,
                    DATALEN,
                    COMPCODE,
                    REASON);

END GET_MSG;

NO_MSG: PROC;
:
END NO_MSG;

```

### **Consultando sobre os atributos de um objeto**

Este exemplo demonstra como usar a chamada MQINQ para consultar sobre os atributos de uma fila

Essa extração não é obtida dos aplicativos de amostra fornecidos com o IBM MQ.

```

%INCLUDE SYSLIB(CMQP);
%INCLUDE SYSLIB(CMQEPP);
:
/*****/
/* WORKING STORAGE DECLARATIONS          */
/*****/
DCL COMPCODE          BINARY FIXED (31);
DCL REASON            BINARY FIXED (31);
DCL HCONN             BINARY FIXED (31);
DCL HOBJ              BINARY FIXED (31);
DCL OPTIONS           BINARY FIXED (31);
DCL SELECTORCOUNT   BINARY FIXED (31);
DCL INTATTRCOUNT   BINARY FIXED (31);
DCL 1 SELECTOR_TABLE,
    3 SELECTORS(5)    BINARY FIXED (31);
DCL 1 INTATTR_TABLE,
    3 INTATTRS(5)    BINARY FIXED (31);
DCL CHARATTRLENGTH   BINARY FIXED (31);
DCL CHARATTRS        CHAR(100);
:

/*****/
/* SET VARIABLES FOR INQUIRE CALL        */
/* INQUIRE ON THE CURRENT QUEUE DEPTH    */
/*****/

        SELECTORS(01) = MQIA_CURRENT_Q_DEPTH;

        SELECTORCOUNT = 1;
        INTATTRCOUNT = 1;

        CHARATTRLENGTH = 0;
/*****/
/*
/* HCONN WAS SET BY PREVIOUS MQCONN REQUEST.          */
/* HOBJ WAS SET BY PREVIOUS MQOPEN REQUEST.          */
/*
/*****/
        CALL MQINQ (HCONN,

```

```

HOBJ,
SELECTORCOUNT,
SELECTORS,
INTATTRCOUNT,
INTATTRS,
CHARATTRLENGTH,
CHARATTRS,
COMPCODE,
REASON);

```

```

/*****/
/* TEST THE COMPLETION CODE OF THE INQUIRE CALL. */
/* IF THE CALL HAS FAILED ISSUE AN ERROR MESSAGE SHOWING */
/* THE COMPLETION CODE AND THE REASON CODE. */
/*****/
IF COMPCODE = MQCC_OK
THEN DO;
:
CALL ERROR_ROUTINE;
END;

```

### **Configurando os atributos de uma fila**

Este exemplo demonstra como usar a chamada MQSET para alterar os atributos de uma fila

Essa extração não é obtida dos aplicativos de amostra fornecidos com o IBM MQ.

```

%INCLUDE SYSLIB(CMQP);
%INCLUDE SYSLIB(CMQEPP);
:
/*****/
/* WORKING STORAGE DECLARATIONS */
/*****/
DCL COMPCODE          BINARY FIXED (31);
DCL REASON            BINARY FIXED (31);
DCL HCONN            BINARY FIXED (31);
DCL HOBJ             BINARY FIXED (31);
DCL OPTIONS          BINARY FIXED (31);
DCL SELECTORCOUNT  BINARY FIXED (31);
DCL INTATTRCOUNT  BINARY FIXED (31);
DCL 1 SELECTOR_TABLE,
   3 SELECTORS(5)    BINARY FIXED (31);
DCL 1 INTATTR_TABLE,
   3 INTATTRS(5)    BINARY FIXED (31);
DCL CHARATTRLENGTH  BINARY FIXED (31);
DCL CHARATTRS       CHAR(100);
:

/*****/
/* SET VARIABLES FOR SET CALL */
/* SET GET AND PUT INHIBITED */
/*****/

SELECTORS(01) = MQIA_INHIBIT_GET;
SELECTORS(02) = MQIA_INHIBIT_PUT;

INTATTRS(01) = MQQA_GET_INHIBITED;
INTATTRS(02) = MQQA_PUT_INHIBITED;

SELECTORCOUNT = 2;
INTATTRCOUNT = 2;

CHARATTRLENGTH = 0;

/*****/
/*
/* HCONN WAS SET BY PREVIOUS MQCONN REQUEST. */
/* HOBJ WAS SET BY PREVIOUS MQOPEN REQUEST. */
/*
/*****/
CALL MQSET (HCONN,
           HOBJ,
           SELECTORCOUNT,
           SELECTORS,
           INTATTRCOUNT,

```



```

        INTATTRS,
        CHARATTRLENGTH,
        CHARATTRS,
        COMPCODE,
        REASON);

/*****
/* TEST THE COMPLETION CODE OF THE SET CALL.          */
/* IF THE CALL HAS FAILED ISSUE AN ERROR MESSAGE SHOWING */
/* THE COMPLETION CODE AND THE REASON CODE.          */
*****/
        IF COMPCODE -/= MQCC_OK
            THEN DO;
            :
            CALL ERROR_ROUTINE;
        END;

```

## Constantes

Utilize as informações de referência desta seção para executar as tarefas que determinam as necessidades de negócios.

### IBM MQ COPY, cabeçalho, inclusão e arquivos de módulo

Essas informações são informações de interface de programação de uso geral

Esta seção contém informações para ajudar a usar o MQI para várias linguagens de programação, conforme a seguir.

#### Arquivos de cabeçalho em C

Os arquivos de cabeçalho são fornecidos para ajudar a gravar programas de aplicativo C que usam o MQI.

Os arquivos de cabeçalho C são resumidos na seguinte tabela:

<i>Tabela 1. Arquivos de cabeçalho C-protótipos de chamada, tipos de dados, códigos de retorno, constantes e estruturas</i>					
Nome do arquivo	Descrição	IBM i	SistemasAI X and Linux®	Windows	z/OS
<b>Chamar Protótipos, Tipos de Dados, Códigos de Retorno, Constantes e Estruturas</b>					
CMQC	definições de MQI	C	C	C	C
CMQBC	Definições MQAI	C	C	C	
CMQEC	Definição de Pontos de Entrada da Interface (inclui CMQC, CMQXC e CMQZC).		C	C	
CMQCFC	definições PCF	C	C	C	C
CMQPSC	Definições de Publicação / Assinatura	C	C	C	C
CMQXC	Definições de canal e saída	C	C	C	C
CMQZC	Definições de serviços instaláveis.	C	C	C	
<b>Chave:</b> C= Arquivos fornecidos					

#### Arquivos de cópia COBOL

Vários arquivos COPY são fornecidos para ajudar a gravar programas de aplicativos COBOL que usam o MQI.

Tabela 2. Arquivos de cópia COBOL-códigos de retorno, constantes e estruturas

Nome do arquivo	Descrição	IBM i	AIX and Linux	Windows	z/OS
<b>Códigos de retorno e constantes</b>					
CMQx	definições de MQI	V	V	V	V
CMQCFx	definições PCF	V	V	V	V
CMQPSx	Definições de Publicação / Assinatura	V	V	V	V
CMQXx	Definições de canal e saída	V	V	V	V
<b>Estruturas</b>					
CMQAIRx	MQAIR-Registro de informações sobre autenticação.		V L	V L	
CMQBOx	MQBO-Opções de início de	V L	V L	V L	
CMQCDx	MQCD-Definição de canal	V L	V L	V L	V L
CMQCFBFx	MQCFBF-parâmetro de filtro de sequência de bytes PCF	V L	V L	V L	V L
CMQCFBSx	MQCFBS-Parâmetro de sequência de bytes PCF	V L	V L	V L	V L
CMQCFGRx	MQCFGR-Parâmetro do grupo PCF	V L	V L	V L	V L
CMQCFHx	MQCFH-Cabeçalho PCF	V L	V L	V L	V L
CMQCFIFx	MQCFIF-Parâmetro de filtro de número inteiro PCF	V L	V L	V L	V L
CMQCFILx	MQCFIL-Parâmetro da lista de inteiros PCF	V L	V L	V L	V L
CMQCFINx	MQCFIN-Parâmetro de número inteiro PCF	V L	V L	V L	V L
CMQCFSFx	MQCFSF-parâmetro de filtro de sequência PCF	V L	V L	V L	V L
CMQCFSLx	MQCFSL-parâmetro da lista de sequências PCF	V L	V L	V L	V L
CMQCFSTx	MQCFST-Parâmetro de sequência PCF	V L	V L	V L	V L
CMQCFXLx	MQCFIL64 -Parâmetro de lista de números inteiros PCF de 64 bits	V L	V L	V L	V L
CMQCFXNx	MQCFIN64 -parâmetro de número inteiro PCF de 64 bits	V L	V L	V L	V L
CMQCHRVx	MQCHARV-Sequência de comprimento variável	V L	V L	V L	V L
CMQCIHx	MQCIH-cabeçalho CICS bridge	V L	V L	V L	V L
CMQCNOx	MQCNO-Opções de conexão	V L	V L	V L	V L
CMQCSPx	MQCSP-Parâmetros de segurança	V L	V L	V L	V L

Tabela 2. Arquivos de cópia COBOL-códigos de retorno, constantes e estruturas (continuação)

Nome do arquivo	Descrição	IBM i	AIX and Linux	Windows	z/OS
CMQCXPx	MQCXP-Parâmetros de saída do canal	V L			V L
CMQDHx	MQDH - Cabeçalho de distribuição	V L	V L	V L	V L
CMQDLHx	MQDLH-Cabeçalho de Devoluções	V L	V L	V L	V L
CMQDXPx	MQDXP-Parâmetros de saída de conversão de dados	V L		V L	
CMQEPHx	MQEPH - cabeçalho PCF integrado	V L	V L	V L	V L
CMQGMOx	MQGMO-Obter opções de mensagem.	V L	V L	V L	V L
CMQIIHx	MQIIH - Informações de cabeçalho do IMS	V L	V L	V L	V L
CMQMDx	MQMD - descritor de mensagem	V L	V L	V L	V L
CMQMD1x	MQMD1 -Descritor de mensagens versão 1	V L	V L	V L	V L
CMQMD2x	MQMD2 -Descritor de mensagens versão 2	V L	V L	V L	V L
CMQMDEx	MQMDE-descritor de mensagens estendido	V L	V L	V L	V L
CMQODx	MQOD-Descritor de objetos	V L	V L	V L	V L
CMQORx	MQOR-Registro do objeto..	V L	V L	V L	V L
CMQPMOx	MQPMO-Opções de mensagem de colocação	V L	V L	V L	V L
CMQRFHx	MQRFH - Regras e formatação de cabeçalho	V L	V L	V L	V L
CMQRFH2x	MQRFH2 - Regras e Formatação do Cabeçalho 2	V L	V L	V L	V L
CMQRMHx	MQRMH - cabeçalho de mensagem de referência	V L	V L	V L	V L
CMQRRx	MQRR-Registro de Resposta.	V L	V L	V L	
CMQSCOx	Opções de configuração MQSCO-TLS		V L	V L	
CMQTMx	MQTM-Mensagem do acionador	V L		V L	V L
CMQTMCx	MQTM-Caractere da mensagem do acionador	V L	V L		
CMQTM2x	MQTM2 -Caractere da mensagem do acionador 2	V L	V L	V L	V L
CMQWIHx	MQWIH - Cabeçalho de informações de trabalho	V L	V L	V L	V L

Tabela 2. Arquivos de cópia COBOL-códigos de retorno, constantes e estruturas (continuação)

Nome do arquivo	Descrição	IBM i	AIX and Linux	Windows	z/OS
CMQXQHx	MQXQH - Cabeçalho da fila de transmissão	V L	V L	V L	V L

**Tecla:**

- Arquivos com valores iniciais fornecidos, x = V
- Arquivos sem valores iniciais fornecidos, x = L

### z/OS Arquivos de inclusão PL/I

Vários arquivos INCLUDE são fornecidos para a linguagem de programação PL/I. Esses arquivos estão disponíveis apenas no z/OS

Tabela 3. Arquivos include PL/I-tipos de dados, códigos de retorno, constantes e estruturas

Nome do arquivo	Descrição	IBM i	AIX and Linux	Windows	z/OS
<b>Tipos de dados, códigos de retorno, constantes e estruturas</b>					
CMQP	definições de MQI				P
CMQCFP	definições PCF				P
CMQEPP	Definições de ponto de entrada:				P
CMQPSP	Definições de Publicação / Assinatura				P
CMQXP	Definições de canal e saída				P

**Chave:** P= Arquivo fornecido.

### IBM i Arquivos de cópia RPG

Os arquivos RPG COPY são fornecidos para a linguagem de programação RPG.. Esses arquivos estão disponíveis apenas em IBM i

Tabela 4. Arquivos de cópia RPG-códigos de retorno, constantes e estruturas

Nome do arquivo	Descrição	IBM i	AIX and Linux	Windows	z/OS
<b>Códigos de retorno e constantes</b>					
CMQx	definições de MQI	G R			
CMQCFx	definições PCF	G			
CMQPSx	Definições de Publicação / Assinatura	G			
CMQXx	Definições de canal e saída	G R			
<b>Estruturas</b>					
CMQBOX	MQBO-Opções de início de	G H			
CMQCDx	MQCD-Definição de canal	G H R			

Tabela 4. Arquivos de cópia RPG-códigos de retorno, constantes e estruturas (continuação)

<b>Nome do arquivo</b>	<b>Descrição</b>	<b>IBM i</b>	<b>AIX and Linux</b>	<b>Windows</b>	<b>z/OS</b>
CMQCFBFx	MQCFBF-parâmetro de filtro de sequência de bytes PCF	G H			
CMQCFBSx	MQCFBS-Parâmetro de sequência de bytes PCF	G H			
CMQCFGRx	MQCFGR-Parâmetro do grupo PCF	G H			
CMQCFHx	MQCFH-Cabeçalho PCF	G H			
CMQCFIFx	MQCFIF-Parâmetro de filtro de número inteiro PCF	G H			
CMQCFILx	MQCFIL-Parâmetro da lista de inteiros PCF	G H			
CMQCFINx	MQCFIN-Parâmetro de número inteiro PCF	G H			
CMQCFSFx	MQCFSF-parâmetro de filtro de sequência PCF	G H			
CMQCFSLx	MQCFSL-parâmetro da lista de sequências PCF	G H			
CMQCFSTx	MQCFST-Parâmetro de sequência PCF	G H			
CMQCFXLx	MQCFIL64 -Parâmetro de lista de números inteiros PCF de 64 bits	G H			
CMQCFXNx	MQCFIN64 -parâmetro de número inteiro PCF de 64 bits	G H			
CMQCHARVx	MQCHARV-Sequência de comprimento variável	G H			
CMQCIHx	MQCIH-cabeçalho CICS bridge	G H			
CMQCNOx	MQCNO-Opções de conexão	G H			
CMQCSPx	MQCSP-Parâmetros de segurança	G H			
CMQCXPx	MQCXP-Parâmetros de saída do canal	G H R			
CMQDHx	MQDH - Cabeçalho de distribuição	G H R			
CMQDLHx	MQDLH-Cabeçalho de Devoluções	G H R			
CMQDXPx	MQDXP-Parâmetros de saída de conversão de dados	G H R			
CMQEPHx	MQEPH - cabeçalho PCF integrado	G H			
CMQGMOx	MQGMO-Obter opções de mensagem.	G H R			
CMQIIHx	MQIIH - Informações de cabeçalho do IMS	G H R			
CMQMDx	MQMD - descritor de mensagem	G H R			

Tabela 4. Arquivos de cópia RPG-códigos de retorno, constantes e estruturas (continuação)

Nome do arquivo	Descrição	IBM i	AIX and Linux	Windows	z/OS
CMQMD1x	MQMD1 -Descritor de mensagens versão 1	G H R			
CMQMD2x	MQMD2 -Descritor de mensagens versão 2	G H			
CMQMDEx	MQMDE-descritor de mensagens estendido	G H R			
CMQODx	MQOD-Descritor de objetos	G H R			
CMQORx	MQOR-Registro do objeto..	G H R			
CMQPMOx	MQPMO-Opções de mensagem de colocação	G H R			
CMQXPx	MQXP-Parâmetros de saída de roteamento de Publicação / Assinatura	G H			
CMQRFHx	MQRFH - Regras e formatação de cabeçalho	G H			
CMQRFH2x	MQRFH2 - Regras e Formatação do Cabeçalho 2	G H			
CMQRMHx	MQRMH - cabeçalho de mensagem de referência	G H R			
CMQRRx	MQRR-Registro de Resposta.	G H R			
CMQTMx	MQTM-Mensagem do acionador	G H R			
CMQTMcx	MQTMc- Caractere da mensagem do acionador	G H R			
CMQTM2x	MQTM2 -Caractere da mensagem do acionador 2	G H R			
CMQWIHx	MQWIH - Cabeçalho de informações de trabalho	G H			
CMQXQHx	MQXQH - Cabeçalho da fila de transmissão	G H R			

**Tecla:**

- Arquivo para ligação estática, inicializado, fornecido x = G
- Arquivo para ligação estática, não inicializado, fornecido x = H
- Arquivo para ligação dinâmica, inicializado, fornecido, x = R

**Windows** **Arquivos de módulo Visual Basic**

Os arquivos de cabeçalho (ou formulário) são fornecidos para ajudá-lo a gravar programas aplicativos do Visual Basic que usam o MQI. Esses arquivos de cabeçalho são fornecidos apenas em versões de 32 bit...

Tabela 5. Arquivos do módulo Visual Basic-declarações de chamada, tipos de dados, códigos de retorno, constantes e estruturas

Nome do arquivo	Descrição	IBM i	SistemasAI X and Linux	Windows	z/OS
<b>Declarações de chamada, tipos de dado, códigos de retorno, constantes e estruturas</b>					
CMQB	definições de MQI			B	
CMQBB	Definições MQAI			B	
CMQCFB	definições PCF			B	
CMQXB	Definições de canal e saída			B	
<b>Chave:</b> B= Arquivo fornecido					

### z/OS z/OS Assembler arquivos COPY

Vários arquivos COPY são fornecidos para ajudar a gravar programas de aplicativo Assembler do z/OS que usam o MQI.

Tabela 6. z/OS Assembler copiar arquivos-tipos de dados, códigos de retorno, constantes e estruturas

Nome do arquivo	Descrição	IBM i	AIX and Linux	Windows	z/OS
<b>Tipos de dados, códigos de retorno e constantes</b>					
CMQA	definições de MQI				A
CMQCFA	definições PCF				A
CMQPSA	Definições de Publicação / Assinatura				A
CMQVERA	Controle de versão da estrutura				A
CMQXA	Definições de canal e saída				A
<b>Estruturas</b>					
CMQCDA	MQCD-Definição de canal				
CMQCFBFA	MQCFBF-parâmetro de filtro de sequência de bytes PCF				
CMQCFBSA	MQCFBS-Parâmetro de sequência de bytes PCF				A
CMQCFGRA	MQCFGR-Parâmetro do grupo PCF				A
CMQCFHA	MQCFH-Cabeçalho PCF				A
CMQCFIFA	MQCFIF-Parâmetro de filtro de número inteiro PCF				A
CMQCFILA	MQCFIL-Parâmetro da lista de inteiros PCF				A
CMQCFINA	MQCFIN-Parâmetro de número inteiro PCF				A
CMQCFSA	MQCFSF-parâmetro de filtro de sequência PCF				A

Tabela 6. z/OS Assembler copiar arquivos-tipos de dados, códigos de retorno, constantes e estruturas (continuação)

Nome do arquivo	Descrição	IBM i	AIX and Linux	Windows	z/OS
CMQCFSLA	MQCFSL-parâmetro da lista de sequências PCF				A
CMQCFSTA	MQCFST-Parâmetro de sequência PCF				A
CMQCFXLA	MQCFIL64 -Parâmetro de lista de números inteiros PCF de 64 bits				A
CMQCFXNA	MQCFIN64 -parâmetro de número inteiro PCF de 64 bits				A
CMQCHARVA	MQCHARV-Sequência de comprimento variável				A
CMQCIHA	MQCIH-cabeçalho CICS bridge				A
CMQCNOA	MQCNO-Opções de conexão				A
CMQCSPA	MQCSP-Parâmetros de segurança				A
CMQCXPA	MQCXP-Parâmetros de saída do canal				A
CMQDHA	MQDH - Cabeçalho de distribuição				A
CMQDLHA	MQDLH-Cabeçalho de Devoluções				A
CMQDXPA	MQDXP-Parâmetros de saída de conversão de dados				A
CMQEPHA	MQEPH - cabeçalho PCF integrado				A
CMQGMOA	MQGMO-Obter opções de mensagem.				A
CMQIIHA	MQIIH - Informações de cabeçalho do IMS				A
CMQMDA	MQMD - descritor de mensagem				A
CMQMD1A	MQMD1 -Descritor de mensagens versão 1				A
CMQMD2A	MQMD2 -Descritor de mensagens versão 2				A
CMQMDEA	MQMDE-descritor de mensagens estendido				A
CMQODA	MQOD-Descritor de objetos				A
CMQORA	MQOR-Registro do objeto..				A
CMQPMOA	MQPMO-Opções de mensagem de colocação				A
CMQRFHA	MQRFH - Regras e formatação de cabeçalho				A



Tabela 6. z/OS Assembler copiar arquivos-tipos de dados, códigos de retorno, constantes e estruturas (continuação)

<b>Nome do arquivo</b>	<b>Descrição</b>	<b>IBM i</b>	<b>AIX and Linux</b>	<b>Windows</b>	<b>z/OS</b>
CMQRFH2A	MQRFH2 - Regras e Formatação do Cabeçalho 2				A
CMQRMHA	MQRMH - cabeçalho de mensagem de referência				A
CMQTMA	MQTM-Mensagem do acionador				A
CMQTMC2A	MQTM2 -Caractere da mensagem do acionador 2				A
CMQWCRA	MQWCR-Registro do cluster de carga de trabalho do cluster.				A
CMQWDRA	MQWDR-Registro de destino de carga de trabalho do cluster				A
CMQWDR1A	MQWDR1 -Registro de destino de carga de trabalho de cluster versão 1				A
CMQWDR2A	MQWDR2 -Registro de destino da carga de trabalho do cluster versão 2				A
CMQWIHA	MQWIH - Cabeçalho de informações de trabalho				A
CMQWQRA	MQWQR-Registro da fila de carga de trabalho do cluster				A
CMQWQR1A	MQWQR1 -Registro da fila de cargas de trabalho do cluster versão 1				A
CMQWQR2A	MQWQR2 -Registro da fila de carga de trabalho do cluster versão 2				A
CMQWXP	MQWXP-Parâmetros de saída de carga de trabalho do cluster				A
CMQWXP1A	MQWXP1 -Parâmetros de saída de carga de trabalho do cluster versão 1				A
CMQWXP2A	MQWXP2 -Parâmetros de saída de carga de trabalho do cluster versão 2				A
CMQWXP3A	MQWXP3 -Parâmetros de saída de carga de trabalho do cluster versão 3				A
CMQXPA	MQXP- CICS API-cruzando parâmetros de saída				A
CMQXQHA	MQXQH - Cabeçalho da fila de transmissão				A

Tabela 6. z/OS Assembler copiar arquivos-tipos de dados, códigos de retorno, constantes e estruturas (continuação)

Nome do arquivo	Descrição	IBM i	AIX and Linux	Windows	z/OS
CMQXWDA	MQXWD-Descriptor de espera de saída				A

**Chave:** A= Arquivo fornecido.

## MQ\_\* (Comprimentos de Sequência).

Tabela 7. Valores de constantes

Nome	Valor decimal	Valor Hexadecimal
MQ_ABEND_CODE_LENGTH	4	X'00000004'
MQ_ACCOUNTING_TOKEN_LENGTH	32	X'00000020'
MQ_APPL_FUNCTION_NAME_LENGTH	10	X'0000000A'
MQ_APPL_IDENTITY_DATA_LENGTH	32	X'00000020'
MQ_APPL_NAME_LENGTH	28	X'0000001C'
MQ_APPL_ORIGIN_DATA_LENGTH	4	X'00000004'
MQ_APPL_TAG_LENGTH	28	X'0000001C'
MQ_ARM_SUFFIX_LENGTH	2	X'00000002'
MQ_ATTENTION_ID_LENGTH	4	X'00000004'
MQ_AUTH_INFO_CONN_NAME_LENGTH	264	X'00000108'
MQ_AUTH_INFO_DESC_LENGTH	64	X'00000040'
MQ_AUTH_INFO_NAME_LENGTH	48	X'00000030'
MQ_AUTH_INFO_OCSP_URL_LENGTH	256	X'00000100'
MQ_AUTHENTICATOR_LENGTH	8	X'00000008'
MQ_AUTO_REORG_CATALOG_LENGTH	44	X'0000002C'
MQ_AUTO_REORG_TIME_LENGTH	4	X'00000004'
MQ_BATCH_INTERFACE_ID_LENGTH	8	X'00000008'
MQ_BRIDGE_NAME_LENGTH	24	X'00000018'
MQ_CANCEL_CODE_LENGTH	4	X'00000004'
MQ_CF_STRUC_DESC_LENGTH	64	X'00000040'
MQ_CF_STRUC_NAME_LENGTH	12	X'0000000C'
MQ_CHANNEL_DATE_LENGTH	12	X'0000000C'
MQ_CHANNEL_DESC_LENGTH	64	X'00000040'
MQ_CHANNEL_NAME_LENGTH	20	X'00000014'
MQ_CHANNEL_TIME_LENGTH	8	X'00000008'
MQ_CHINIT_SERVICE_PARM_LENGTH	32	X'00000020'
MQ_CICS_FILE_NAME_LENGTH	8	X'00000008'
MQ_CLIENT_ID_LENGTH	23	X'00000017'
MQ_CLUSTER_NAME_LENGTH	48	X'00000030'
MQ_CONN_NAME_LENGTH	264	X'00000108'

<i>Tabela 7. Valores de constantes (continuação)</i>		
<b>Nome</b>	<b>Valor decimal</b>	<b>Valor Hexadecimal</b>
MQ_CONN_TAG_LENGTH	128	X'00000080'
MQ_CONNECTION_ID_LENGTH	24	X'00000018'
MQ_CORREL_ID_LENGTH	24	X'00000018'
MQ_CREATION_DATE_LENGTH	12	X'0000000C'
MQ_CREATION_TIME_LENGTH	8	X'00000008'
MQ_DATE_LENGTH	12	X'0000000C'
MQ_DISTINGUISHED_NAME_LENGTH	1024	X'00000400'
MQ_DNS_GROUP_NAME_LENGTH	18	X'00000012'
MQ_EXIT_DATA_LENGTH	32	X'00000020'
MQ_EXIT_INFO_NAME_LENGTH	48	X'00000030'
MQ_EXIT_NAME_LENGTH	(value differs by platform or version)	
MQ_EXIT_PD_AREA_LENGTH	48	X'00000030'
MQ_EXIT_USER_AREA_LENGTH	16	X'00000010'
MQ_FACILITY_LENGTH	8	X'00000008'
MQ_FACILITY_LIKE_LENGTH	4	X'00000004'
MQ_FORMAT_LENGTH	8	X'00000008'
MQ_FUNCTION_LENGTH	4	X'00000004'
MQ_GROUP_ID_LENGTH	24	X'00000018'
MQ_LDAP_PASSWORD_LENGTH	32	X'00000020'
MQ_LISTENER_NAME_LENGTH	48	X'00000030'
MQ_LISTENER_DESC_LENGTH	64	X'00000040'
MQ_LOCAL_ADDRESS_LENGTH	48	X'00000030'
MQ_LTERM_OVERRIDE_LENGTH	8	X'00000008'
MQ_LU_NAME_LENGTH	8	X'00000008'
MQ_LUWID_LENGTH	16	X'00000010'
MQ_MAX_EXIT_NAME_LENGTH	128	X'00000080'
MQ_MAX_MCA_USER_ID_LENGTH	64	X'00000040'
MQ_MAX_PROPERTY_NAME_LENGTH	4095	X'0000FFFF'
MQ_MAX_USER_ID_LENGTH	64	X'00000040'
MQ_MCA_JOB_NAME_LENGTH	28	X'0000001C'
MQ_MCA_NAME_LENGTH	20	X'00000014'
MQ_MCA_USER_DATA_LENGTH	32	X'00000020'
MQ_MCA_USER_ID_LENGTH	(value differs by platform or version)	
MQ_MFS_MAP_NAME_LENGTH	8	X'00000008'
MQ_MODE_NAME_LENGTH	8	X'00000008'
MQ_MSG_HEADER_LENGTH	4000	X'00000FA0'
MQ_MSG_ID_LENGTH	24	X'00000018'
MQ_MSG_TOKEN_LENGTH	16	X'00000010'
MQ_NAMELIST_DESC_LENGTH	64	X'00000040'

<i>Tabela 7. Valores de constantes (continuação)</i>		
<b>Nome</b>	<b>Valor decimal</b>	<b>Valor Hexadecimal</b>
MQ_NAMELIST_NAME_LENGTH	48	X'00000030'
MQ_NHA_INSTANCE_NAME_LENGTH	48	X'00000030'
MQ_OBJECT_INSTANCE_ID_LENGTH	24	X'00000018'
MQ_OBJECT_NAME_LENGTH	48	X'00000030'
MQ_PASS_TICKET_APPL_LENGTH	8	X'00000008'
MQ_PASSWORD_LENGTH	12	X'0000000C'
MQ_PROCESS_APPL_ID_LENGTH	256	X'00000100'
MQ_PROCESS_DESC_LENGTH	64	X'00000040'
MQ_PROCESS_ENV_DATA_LENGTH	128	X'00000080'
MQ_PROCESS_NAME_LENGTH	48	X'00000030'
MQ_PROCESS_USER_DATA_LENGTH	128	X'00000080'
MQ_PROGRAM_NAME_LENGTH	20	X'00000014'
MQ_PUT_APPL_NAME_LENGTH	28	X'0000001C'
MQ_PUT_DATE_LENGTH	8	X'00000008'
MQ_PUT_TIME_LENGTH	8	X'00000008'
MQ_Q_DESC_LENGTH	64	X'00000040'
MQ_Q_MGR_DESC_LENGTH	64	X'00000040'
MQ_Q_MGR_IDENTIFIER_LENGTH	48	X'00000030'
MQ_Q_MGR_NAME_LENGTH	48	X'00000030'
MQ_Q_NAME_LENGTH	48	X'00000030'
MQ_QSG_NAME_LENGTH	4	X'00000004'
MQ_REMOTE_SYS_ID_LENGTH	4	X'00000004'
MQ_SECURITY_ID_LENGTH	40	X'00000028'
MQ_SELECTOR_LENGTH	10240	X'00002800'
MQ_SERVICE_ARGS_LENGTH	255	X'000000FF'
MQ_SERVICE_COMMAND_LENGTH	255	X'000000FF'
MQ_SERVICE_DESC_LENGTH	64	X'00000040'
MQ_SERVICE_NAME_LENGTH	32	X'00000020'
MQ_SERVICE_PATH_LENGTH	255	X'000000FF'
MQ_SERVICE_STEP_LENGTH	8	X'00000008'
MQ_SHORT_CONN_NAME_LENGTH	20	X'00000014'
MQ_SHORT_DNAME_LENGTH	256	X'00000100'
MQ_SSL_CIPHER_SPEC_LENGTH	32	X'00000020'
MQ_SSL_CRYPTOHARDWARE_LENGTH	256	X'00000100'
MQ_SSL_HANDSHAKE_STAGE_LENGTH	32	X'00000020'
MQ_SSL_KEY_LIBRARY_LENGTH	44	X'0000002C'
MQ_SSL_KEY_MEMBER_LENGTH	8	X'00000008'
MQ_SSL_KEY_REPOSITORY_LENGTH	256	X'00000100'
MQ_SSL_PEER_NAME_LENGTH	1024	X'00000400'

<i>Tabela 7. Valores de constantes (continuação)</i>		
<b>Nome</b>	<b>Valor decimal</b>	<b>Valor Hexadecimal</b>
MQ_SSL_SHORT_PEER_NAME_LENGTH	256	X'00000100'
MQ_START_CODE_LENGTH	4	X'00000004'
MQ_STORAGE_CLASS_DESC_LENGTH	64	X'00000040'
MQ_STORAGE_CLASS_LENGTH	8	X'00000008'
MQ_SUB_IDENTITY_LENGTH	128	X'00000080'
MQ_SUB_POINT_LENGTH	128	X'00000080'
MQ_SUITE_B_128_BIT	2	X'00000002'
MQ_SUITE_B_192_BIT	4	X'00000004'
MQ_SUITE_B_NONE	1	X'00000001'
MQ_SUITE_B_NOT_AVAILABLE	0	X'00000000'
MQ_TCP_NAME_LENGTH	8	X'00000008'
MQ_TIME_LENGTH	8	X'00000008'
MQ_TOPIC_DESC_LENGTH	64	X'00000040'
MQ_TOPIC_NAME_LENGTH	48	X'00000030'
MQ_TOPIC_STR_LENGTH	10240	X'00002800'
MQ_TOTAL_EXIT_DATA_LENGTH	999	X'000003E7'
MQ_TOTAL_EXIT_NAME_LENGTH	999	X'000003E7'
MQ_TP_NAME_LENGTH	64	X'00000040'
MQ_TPIPE_NAME_LENGTH	8	X'00000008'
MQ_TRAN_INSTANCE_ID_LENGTH	16	X'00000010'
MQ_TRANSACTION_ID_LENGTH	4	X'00000004'
MQ_TRIGGER_DATA_LENGTH	64	X'00000040'
MQ_TRIGGER_PROGRAM_NAME_LENGTH	8	X'00000008'
MQ_TRIGGER_TERM_ID_LENGTH	4	X'00000004'
MQ_TRIGGER_TRANS_ID_LENGTH	4	X'00000004'
MQ_USER_ID_LENGTH	12	X'0000000C'
MQ_VERSION_LENGTH	8	X'00000008'
MQ_XCF_GROUP_NAME_LENGTH	8	X'00000008'
MQ_XCF_MEMBER_NAME_LENGTH	16	X'00000010'

### ***MQ\_ \* (Comprimentos de Sequência de Formato de Comando)***

<i>Tabela 8. Valores de constantes</i>		
<b>Nome</b>	<b>Valor decimal</b>	<b>Valor Hexadecimal</b>
MQ_ARCHIVE_PFX_LENGTH	36	X'00000024'
MQ_ARCHIVE_UNIT_LENGTH	8	X'00000008'
MQ_ASID_LENGTH	4	X'00000004'
MQ_AUTH_PROFILE_NAME_LENGTH	48	X'00000030'
MQ_CF_LEID_LENGTH	12	X'0000000C'
MQ_COMMAND_MQSC_LENGTH	32768	X'00008000'
MQ_DATA_SET_NAME_LENGTH	44	X'0000002C'

<i>Tabela 8. Valores de constantes (continuação)</i>		
<b>Nome</b>	<b>Valor decimal</b>	<b>Valor Hexadecimal</b>
MQ_DB2_NAME_LENGTH	4	X'00000004'
MQ_DSG_NAME_LENGTH	8	X'00000008'
MQ_ENTITY_NAME_LENGTH	1024	X'00000400'
MQ_ENV_INFO_LENGTH	96	X'00000060'
MQ_IP_ADDRESS_LENGTH	48	X'00000030'
MQ_LOG_CORREL_ID_LENGTH	8	X'00000008'
MQ_LOG_EXTENT_NAME_LENGTH	24	X'00000018'
MQ_LOG_PATH_LENGTH	1024	X'00000400'
MQ_LRSN_LENGTH	12	X'0000000C'
MQ_ORIGIN_NAME_LENGTH	8	X'00000008'
MQ_PSB_NAME_LENGTH	8	X'00000008'
MQ_PST_ID_LENGTH	8	X'00000008'
MQ_Q_MGR_CPF_LENGTH	4	X'00000004'
MQ_RESPONSE_ID_LENGTH	24	X'00000018'
MQ_RBA_LENGTH	16	X'00000010'
MQ_SECURITY_PROFILE_LENGTH	40	X'00000028'
MQ_SERVICE_COMPONENT_LENGTH	48	X'00000030'
MQ_SUB_NAME_LENGTH	10240	X'00002800'
MQ_SYSP_SERVICE_LENGTH	32	X'00000020'
MQ_SYSTEM_NAME_LENGTH	8	X'00000008'
MQ_TASK_NUMBER_LENGTH	8	X'00000008'
MQ_TPIPE_PFX_LENGTH	4	X'00000004'
MQ_UOW_ID_LENGTH	256	X'00000100'
MQ_USER_DATA_LENGTH	10240	X'00002800'
MQ_VOLSER_LENGTH	6	X'00000006'

### **MQACH\_\* (estrutura do cabeçalho da área da sequência de saída da API).**

<i>Tabela 9. Estruturas de constantes</i>	
<b>Nome</b>	<b>Estrutura</b>
MQACH_STRUC_ID	"ACH~"
MQACH_STRUC_ID_ARRAY	'A','C','H','~'

**Nota:** O símbolo ~ representa um único caractere em branco.

<i>Tabela 10. Valores de constantes</i>		
<b>Nome</b>	<b>Valor decimal</b>	<b>Valor Hexadecimal</b>
MQACH_VERSION_1	1	X'00000001'
MQACH_CURRENT_VERSION	1	X'00000001'
MQACH_LENGTH_1	(value differs by platform or version)	(value differs by platform or version)
MQACH_CURRENT_LENGTH	(value differs by platform or version)	(value differs by platform or version)

## MQACT\_\* (Token de Contabilidade)

Tabela 11. Nomes e valores de constantes	
Nome	Value
MQACT_NONE	X'00...00' (32 nulos)
MQACT_NONE_ARRAY	'\0', '\0', ... (32 nulos)

## MQACT\_\* (Opções de Ação do Formato de Comandos)..

Tabela 12. Valores de constantes		
Nome	Valor decimal	Valor Hexadecimal
MQACT_FORCE_REMOVE	1	X'00000001'
MQACT_ADVANCE_LOG	2	X'00000002'
MQACT_COLLECT_STATISTICS	3	X'00000003'
MQACT_PUBSUB	4	X'00000004'

## MQACTP\_\* (Ação)

Tabela 13. Valores de constantes		
Nome	Valor decimal	Valor Hexadecimal
MQACTP_NEW	0	X'00000000'
MQACTP_FORWARD	1	X'00000001'
MQACTP_REPLY	2	X'00000002'
MQACTP_REPORT	3	X'00000003'

## MQACTT\_\* (Tipos de Token de Contabilidade).

Tabela 14. Valores de constantes	
Nome	Valor Hexadecimal
MQACTT_UNKNOWN	X'00'
MQACTT_CICS_LUOW_ID	X'01'
MQACTT_OS2_DEFAULT	X'04'
MQACTT_DOS_DEFAULT	X'05'
MQACTT_UNIX_NUMERIC_ID	X'06'
MQACTT_OS400_ACCOUNT_TOKEN	X'08'
MQACTT_WINDOWS_DEFAULT	X'09'
MQACTT_NT_SECURITY_ID	X'0B'
MQACTT_USER	X'19'

## MQADOPT\_\* (adotar novas verificações de MCA e adotar novos tipos de MCA)

### Adotar novas verificações de MCA

Tabela 15. Valores de constantes		
Nome	Valor decimal	Valor Hexadecimal
MQADOPT_CHECK_NONE	0	X'00000000'

<i>Tabela 15. Valores de constantes (continuação)</i>		
Nome	Valor decimal	Valor Hexadecimal
MQADOPT_CHECK_ALL	1	X'00000001'
MQADOPT_CHECK_Q_MGR_NAME	2	X'00000002'
MQADOPT_CHECK_NET_ADDR	4	X'00000004'

### Adotar novos tipos de MCA

<i>Tabela 16. Valores de constantes</i>		
Nome	Valor decimal	Valor Hexadecimal
MQADOPT_TYPE_NO	0	X'00000000'
MQADOPT_TYPE_ALL	1	X'00000001'
MQADOPT_TYPE_SVR	2	X'00000002'
MQADOPT_TYPE_SDR	4	X'00000004'
MQADOPT_TYPE_RCVR	8	X'00000008'
MQADOPT_TYPE_CLUSRCVR	16	X'00000010'

### **MQAIR\_\*** (Estrutura do registro de informações de autenticação).

<i>Tabela 17. Estruturas de constantes</i>	
Nome	Estrutura
MQAIR_STRUC_ID	"AIR↵"
MQAIR_STRUC_ID_ARRAY	'A', 'I', 'R', '↵'

**Nota:** O símbolo ↵ representa um único caractere em branco.

<i>Tabela 18. Valores de constantes</i>		
Nome	Valor decimal	Valor Hexadecimal
MQAIR_VERSION_1	1	X'00000001'
MQAIR_VERSION_2	2	X'00000002'
MQAIR_CURRENT_VERSION	2	X'00000002'

### **MQAIT\_\*** (Tipo de Informações sobre Autenticação).

<i>Tabela 19. Valores de constantes</i>		
Nome	Valor decimal	Valor Hexadecimal
MQAIT_ALL	0	X'00000000'
MQAIT_CRL_LDAP	1	X'00000001'
MQAIT_OCSP	2	X'00000002'
MQAIT_IDPW_OS	3	X'00000003'
MQAIT_IDPW_LDAP	4	X'00000004'

### **MQAS\_\*** (Valores de Estado Assíncronos do Formato de Comandos).

<i>Tabela 20. Valores de constantes</i>		
Nome	Valor decimal	Valor Hexadecimal
MQAS_NONE	0	X'00000000'



*Tabela 20. Valores de constantes (continuação)*

Nome	Valor decimal	Valor Hexadecimal
MQAS_STARTED	1	X'00000001'
MQAS_START_WAIT	2	X'00000002'
MQAS_STOPPED	3	X'00000003'
MQAS_SUSPENDED	4	X'00000004'
MQAS_SUSPENDED_TEMPORARY	5	X'00000005'
MQAS_ACTIVE	6	X'00000006'
MQAS_INACTIVE	7	X'00000007'

## MQAT\_\* (Colocar Tipos de Aplicativo)

*Tabela 21. Valores de constantes*

Nome	Valor decimal	Valor Hexadecimal
MQAT_UNKNOWN	-1	X'FFFFFFFF'
MQAT_NO_CONTEXT	0	X'00000000'
MQAT_CICS	1	X'00000001'
MQAT_MVS	2	X'00000002'
MQAT_OS390	2	X'00000002'
MQAT_ZOS	2	X'00000002'
MQAT_IMS	3	X'00000003'
MQAT_OS2	4	X'00000004'
MQAT_DOS	5	X'00000005'
MQAT_AIX	6	X'00000006'
MQAT_UNIX	6	X'00000006'
MQAT_QMGR	7	X'00000007'
MQAT_OS400	8	X'00000008'
MQAT_WINDOWS	9	X'00000009'
MQAT_CICS_VSE	10	X'0000000A'
MQAT_WINDOWS_NT	11	X'0000000B'
MQAT_VMS	12	X'0000000C'
MQAT_GUARDIAN	13	X'0000000D'
MQAT_NSK	13	X'0000000D'
MQAT_VOS	14	X'0000000E'
MQAT_OPEN_TP1	15	X'0000000F'
MQAT_VM	18	X'00000012'
MQAT_IMS_BRIDGE	19	X'00000013'
MQAT_XCF	20	X'00000014'
MQAT_CICS_BRIDGE	21	X'00000015'
MQAT_NOTES_AGENT	22	X'00000016'
MQAT_TPF	23	X'00000017'
MQAT_USER	25	X'00000019'

<i>Tabela 21. Valores de constantes (continuação)</i>		
<b>Nome</b>	<b>Valor decimal</b>	<b>Valor Hexadecimal</b>
MQAT_BROKER	26	X'0000001A'
MQAT_QMGR_PUBLISH	26	X'0000001A'
MQAT_JAVA	28	X'0000001C'
MQAT_DQM	29	X'0000001D'
MQAT_CHANNEL_INITIATOR	30	X'0000001E'
MQAT_WLM	31	X'0000001F'
MQAT_BATCH	32	X'00000020'
MQAT_RRS_BATCH	33	X'00000021'
MQAT_SIB	34	X'00000022'
MQAT_DEFAULT	(value differs by platform or version)	(value differs by platform or version)
MQAT_USER_FIRST	65536	X'00010000'
MQAT_USER_LAST	99999999	X'3B9AC9FF'

### **MQAUTH\_\* (Valores de autoridade de formato de comando)..**

<i>Tabela 22. Valores de constantes</i>		
<b>Nome</b>	<b>Valor decimal</b>	<b>Valor Hexadecimal</b>
MQAUTH_NONE	0	X'00000000'
MQAUTH_ALT_USER_AUTHORITY	1	X'00000001'
MQAUTH_BROWSE	2	X'00000002'
MQAUTH_CHANGE	3	X'00000003'
MQAUTH_CLEAR	4	X'00000004'
MQAUTH_CONNECT	5	X'00000005'
MQAUTH_CREATE	6	X'00000006'
MQAUTH_DELETE	7	X'00000007'
MQAUTH_DISPLAY	8	X'00000008'
MQAUTH_INPUT	9	X'00000009'
MQAUTH_INQUIRE	10	X'0000000A'
MQAUTH_OUTPUT	11	X'0000000B'
MQAUTH_PASS_ALL_CONTEXT	12	X'0000000C'
MQAUTH_PASS_IDENTITY_CONTEXT	13	X'0000000D'
MQAUTH_SET	14	X'0000000E'
MQAUTH_SET_ALL_CONTEXT	15	X'0000000F'
MQAUTH_SET_IDENTITY_CONTEXT	16	X'00000010'
MQAUTH_CONTROL	17	X'00000011'
MQAUTH_CONTROL_EXTENDED	18	X'00000012'
MQAUTH_PUBLISH	19	X'00000013'
MQAUTH_SUBSCRIBE	20	X'00000014'
MQAUTH_RESUME	21	X'00000015'
MQAUTH_SYSTEM	22	X'00000016'

## MQAUTHOPT\_\* (Opções de Autoridade de Formato de Comando).

Tabela 23. Valores de constantes		
Nome	Valor decimal	Valor Hexadecimal
MQAUTHOPT_UMULATIVO	256	X'00000100'
MQAUTHOPT_ENTITY_EXPLICIT	1	X'00000001'
MQAUTHOPT_ENTITY_SET	2	X'00000002'
MQAUTHOPT_NAME_ALL_MATCHING	32	X'00000020'
MQAUTHOPT_NAME_AS_WILDCARD	64	X'00000040'
MQAUTHOPT_NAME_EXPLICIT	16	X'00000010'

## MQAXC\_\* (estrutura de contexto de saída da API).

Tabela 24. Estruturas de constantes	
Nome	Estrutura
MQAXC_STRUC_ID	"AXC↵"
MQAXC_STRUC_ID_ARRAY	'A', 'X', 'C', '↵'

**Nota:** O símbolo ↵ representa um único caractere em branco.

Tabela 25. Valores de constantes		
Nome	Valor decimal	Valor Hexadecimal
MQAXC_VERSION_1	1	X'00000001'
MQAXC_CURRENT_VERSION	1	X'00000001'

## MQAXP\_\* (estrutura do parâmetro de saída de API).

Tabela 26. Estruturas de constantes	
Nome	Estrutura
MQAXP_STRUC_ID	"AXP↵"
MQAXP_STRUC_ID_ARRAY	'A', 'X', 'P', '↵'

**Nota:** O símbolo ↵ representa um único caractere em branco.

Tabela 27. Valores de constantes		
Nome	Valor decimal	Valor Hexadecimal
MQAXP_VERSION_1	1	X'00000001'
MQAXP_VERSION_2	2	X'00000002'
MQAXP_CURRENT_VERSION	2	X'00000002'

## MQBA\_\* (Seletores de atributo de byte)

Tabela 28. Valores de constantes		
Nome	Valor decimal	Valor Hexadecimal
MQBA_FIRST	6001	X'00001771'
MQBA_LAST	8000	X'00001F40'

## MQBACF\_\* (Tipos de Parâmetros de Bytes do Formato de Comando)

*Tabela 29. Valores de constantes*

Nome	Valor decimal	Valor Hexadecimal
MQBACF_FIRST	7001	X'00001B59'
MQBACF_EVENT_ACCOUNTING_TOKEN	7001	X'00001B59'
MQBACF_EVENT_SECURITY_ID	7002	X'00001B5A'
MQBACF_RESPONSE_SET	7003	X'00001B5B'
MQBACF_RESPONSE_ID	7004	X'00001B5C'
MQBACF_EXTERNAL_UOW_ID	7005	X'00001B5D'
MQBACF_CONNECTION_ID	7006	X'00001B5E'
MQBACF_GENERIC_CONNECTION_ID	7007	X'00001B5F'
MQBACF_ORIGIN_UOW_ID	7008	X'00001B60'
MQBACF_Q_MGR_UOW_ID	7009	X'00001B61'
MQBACF_ACCOUNTING_TOKEN	7010	X'00001B62'
MQBACF_CORREL_ID	7011	X'00001B63'
MQBACF_GROUP_ID	7012	X'00001B64'
MQBACF_MSG_ID	7013	X'00001B65'
MQBACF_CF_LEID	7014	X'00001B66'
MQBACF_DESTINATION_CORREL_ID	7015	X'00001B67'
MQBACF_SUB_ID	7016	X'00001B68'
MQBACF_LAST_USED	7016	X'00001B68'

## MQBL\_\* (Comprimento do Buffer para Sequência mqAdde Sequência mqSet)

*Tabela 30. Valores de constantes*

Nome	Valor decimal	Valor Hexadecimal
MQBL_NULL_TERMINATED	-1	X'FFFFFFFF'

## MQBMHO\_\* (Buffer para opções de manipulação de mensagem e estrutura).

### Buffer para estrutura de opções de manipulação de mensagens

*Tabela 31. Estruturas de constantes*

Nome	Estrutura
MQBMHO_STRUC_ID	"BMHO"
MQBMHO_STRUC_ID_ARRAY	'B', 'M', 'H', 'O'

**Nota:** O símbolo ~ representa um único caractere em branco.

*Tabela 32. Valores de constantes*

Nome	Valor decimal	Valor Hexadecimal
MQBMHO_VERSION_1	1	X'00000001'
MQBMHO_CURRENT_VERSION	1	X'00000001'

## Opções de buffer para identificador de mensagem

*Tabela 33. Valores de constantes*

Nome	Valor decimal	Valor Hexadecimal
MQBMHO_NONE	0	X'00000000'
MQBMHO_DELETE_PROPERTIES	1	X'00000001'

## MQBND\_\* (Ligações Padrão)

*Tabela 34. Valores de constantes*

Nome	Valor decimal	Valor Hexadecimal
MQBND_BIND_ON_OPEN	0	X'00000000'
MQBND_BIND_NOT_FIXED	1	X'00000001'
MQBND_BIND_ON_GROUP	2	X'00000002'

## MQBO\_\* (Iniciar opções e estrutura)

### Iniciar estrutura de opções

*Tabela 35. Estruturas de constantes*

Nome	Estrutura
MQBO_STRUC_ID	"B0↵"
MQBO_STRUC_ID_ARRAY	'B', '0', '↵', '↵'

**Nota:** O símbolo ↵ representa um único caractere em branco.

*Tabela 36. Valores de constantes*

Nome	Valor decimal	Valor Hexadecimal
MQBO_VERSION_1	1	X'00000001'
MQBO_CURRENT_VERSION	1	X'00000001'

### Iniciar Opções

*Tabela 37. Valores de constantes*

Nome	Valor decimal	Valor Hexadecimal
MQBO_NONE	0	X'00000000'

## MQBT\_\* (Tipos de ponte de formato de comando)

*Tabela 38. Valores de constantes*

Nome	Valor decimal	Valor Hexadecimal
MQBT_OTMA	1	X'00000001'

## MQCA\_\* (Seletores de atributo de caractere)

*Tabela 39. Valores de constantes*

Nome	Valor decimal	Valor Hexadecimal
MQCA_ADMIN_TOPIC_NAME	2105	X'00000839'
MQCA_ALTERATION_DATE	2027	X'000007EB'

Tabela 39. Valores de constantes (continuação)

Nome	Valor decimal	Valor Hexadecimal
MQCA_ALTERATION_TIME	2028	X'000007EC'
MQCA_APPL_ID	2001	X'000007D1'
MQCA_AUTH_INFO_CONN_NAME	2053	X'00000805'
MQCA_AUTH_INFO_DESC	2046	X'000007FE'
MQCA_AUTH_INFO_NAME	2045	X'000007FD'
MQCA_AUTH_INFO_OCSP_URL	2109	X'0000083D'
MQCA_AUTO_REORG_CATALOG	2091	X'0000082B'
MQCA_AUTO_REORG_START_TIME	2090	X'0000082A'
MQCA_BACKOUT_REQ_Q_NAME	2019	X'000007E3'
MQCA_BASE_OBJECT_NAME	2002	X'000007D2'
MQCA_BASE_Q_NAME	2002	X'000007D2'
MQCA_BATCH_INTERFACE_ID	2068	X'00000814'
MQCA_CF_STRUC_DESC	2052	X'00000804'
MQCA_CF_STRUC_NAME	2039	X'000007F7'
MQCA_CHANNEL_AUTO_DEF_EXIT	2026	X'000007EA'
MQCA_CHILD	2101	X'00000835'
MQCA_CHINIT_SERVICE_PARM	2076	X'0000081C'
MQCA_CICS_FILE_NAME	2060	X'0000080C'
MQCA_CLUS_CHL_NAME	2124	X'0000084C'
MQCA_CLUSTER_DATE	2037	X'000007F5'
MQCA_CLUSTER_NAME	2029	X'000007ED'
MQCA_CLUSTER_NAMELIST	2030	X'000007EE'
MQCA_CLUSTER_Q_MGR_NAME	2031	X'000007EF'
MQCA_CLUSTER_TIME	2038	X'000007F6'
MQCA_CLUSTER_WORKLOAD_DATA	2034	X'000007F2'
MQCA_CLUSTER_WORKLOAD_EXIT	2033	X'000007F1'
MQCA_COMMAND_INPUT_Q_NAME	2003	X'000007D3'
MQCA_COMMAND_REPLY_Q_NAME	2067	X'00000813'
MQCA_CREATION_DATE	2004	X'000007D4'
MQCA_CREATION_TIME	2005	X'000007D5'
MQCA_DEAD_LETTER_Q_NAME	2006	X'000007D6'
MQCA_DEF_XMIT_Q_NAME	2025	X'000007E9'
MQCA_DNS_GROUP	2071	X'00000817'
MQCA_ENV_DATA	2007	X'000007D7'
MQCA_FIRST	2001	X'000007D1'
MQCA_IGQ_USER_ID	2041	X'000007F9'
MQCA_INITIATION_Q_NAME	2008	X'000007D8'
MQCA_LAST	4000	X'00000FA0'
MQCA_LAST_USED	2109	X'0000083D'

Tabela 39. Valores de constantes (continuação)

Nome	Valor decimal	Valor Hexadecimal
MQCA_LDAP_PASSWORD	2048	X'00000800'
MQCA_LDAP_USER_NAME	2047	X'000007FF'
MQCA_LU_GROUP_NAME	2072	X'00000818'
MQCA_LU_NAME	2073	X'00000819'
MQCA_LU62_ARM_SUFFIX	2074	X'0000081A'
MQCA_MODEL_DURABLE_Q	2096	X'00000830'
MQCA_MODEL_NON_DURABLE_Q	2097	X'00000831'
MQCA_MONITOR_Q_NAME	2066	X'00000812'
MQCA_NAMELIST_DESC	2009	X'000007D9'
MQCA_NAMELIST_NAME	2010	X'000007DA'
MQCA_NAMES	2020	X'000007E4'
MQCA_PARENT	2102	X'00000836'
MQCA_PASS_TICKET_APPL	2086	X'00000826'
MQCA_PROCESS_DESC	2011	X'000007DB'
MQCA_PROCESS_NAME	2012	X'000007DC'
MQCA_Q_DESC	2013	X'000007DD'
MQCA_Q_MGR_DESC	2014	X'000007DE'
MQCA_Q_MGR_IDENTIFIER	2032	X'000007F0'
MQCA_Q_MGR_NAME	2015	X'000007DF'
MQCA_Q_NAME	2016	X'000007E0'
MQCA_QSG_NAME	2040	X'000007F8'
MQCA_REMOTE_Q_MGR_NAME	2017	X'000007E1'
MQCA_REMOTE_Q_NAME	2018	X'000007E2'
MQCA_REPOSITORY_NAME	2035	X'000007F3'
MQCA_REPOSITORY_NAMELIST	2036	X'000007F4'
MQCA_RESUME_DATE	2098	X'00000832'
MQCA_RESUME_TIME	2099	X'00000833'
MQCA_SERVICE_DESC	2078	X'0000081E'
MQCA_SERVICE_NAME	2077	X'0000081D'
MQCA_SERVICE_START_ARGS	2080	X'00000820'
MQCA_SERVICE_START_COMMAND	2079	X'0000081F'
MQCA_SERVICE_STOP_ARGS	2082	X'00000822'
MQCA_SERVICE_STOP_COMMAND	2081	X'00000821'
MQCA_STDERR_DESTINATION	2084	X'00000824'
MQCA_STDOUT_DESTINATION	2083	X'00000823'
MQCA_SSL_CRL_NAMELIST	2050	X'00000802'
MQCA_SSL_CRYPTO_HARDWARE	2051	X'00000803'
MQCA_SSL_KEY_LIBRARY	2069	X'00000815'
MQCA_SSL_KEY_MEMBER	2070	X'00000816'

*Tabela 39. Valores de constantes (continuação)*

<b>Nome</b>	<b>Valor decimal</b>	<b>Valor Hexadecimal</b>
MQCA_SSL_KEY_REPOSITORY	2049	X'00000801'
MQCA_STORAGE_CLASS	2022	X'000007E6'
MQCA_STORAGE_CLASS_DESC	2042	X'000007FA'
MQCA_SYSTEM_LOG_Q_NAME	2065	X'00000811'
MQCA_TCP_NAME	2075	X'0000081B'
MQCA_TOPIC_DESC	2093	X'0000082D'
MQCA_TOPIC_NAME	2092	X'0000082C'
MQCA_TOPIC_STRING_FILTER	2108	X'0000083C'
MQCA_TOPIC_STRING	2094	X'0000082E'
MQCA_TPIPE_NAME	2085	X'00000825'
MQCA_TRIGGER_CHANNEL_NAME	2064	X'00000810'
MQCA_TRIGGER_DATA	2023	X'000007E7'
MQCA_TRIGGER_PROGRAM_NAME	2062	X'0000080E'
MQCA_TRIGGER_TERM_ID	2063	X'0000080F'
MQCA_TRIGGER_TRANS_ID	2061	X'0000080D'
MQCA_USER_DATA	2021	X'000007E5'
MQCA_USER_LIST	4000	X'00000FA0'
MQCA_VERSION	2120	X'00000848'
MQCA_XCF_GROUP_NAME	2043	X'000007FB'
MQCA_XCF_MEMBER_NAME	2044	X'000007FC'
MQCA_XMIT_Q_NAME	2024	X'000007E8'

### **MQCACF\_\* (Tipos de Parâmetros de Caracteres do Formato de Comando)**

*Tabela 40. Valores de constantes*

<b>Nome</b>	<b>Valor decimal</b>	<b>Valor Hexadecimal</b>
MQCACF_FIRST	3001	X'00000BB9'
MQCACF_FROM_Q_NAME	3001	X'00000BB9'
MQCACF_TO_Q_NAME	3002	X'00000BBA'
MQCACF_FROM_PROCESS_NAME	3003	X'00000BBB'
MQCACF_TO_PROCESS_NAME	3004	X'00000BBC'
MQCACF_FROM_NAMELIST_NAME	3005	X'00000BBD'
MQCACF_TO_NAMELIST_NAME	3006	X'00000BBE'
MQCACF_FROM_CHANNEL_NAME	3007	X'00000BBF'
MQCACF_TO_CHANNEL_NAME	3008	X'00000BC0'
MQCACF_FROM_AUTH_INFO_NAME	3009	X'00000BC1'
MQCACF_TO_AUTH_INFO_NAME	3010	X'00000BC2'
MQCACF_Q_NAMES	3011	X'00000BC3'
MQCACF_PROCESS_NAMES	3012	X'00000BC4'
MQCACF_NAMELIST_NAMES	3013	X'00000BC5'



Tabela 40. Valores de constantes (continuação)

Nome	Valor decimal	Valor Hexadecimal
MQCACF_ESCAPE_TEXT	3014	X'00000BC6'
MQCACF_LOCAL_Q_NAMES	3015	X'00000BC7'
MQCACF_MODEL_Q_NAMES	3016	X'00000BC8'
MQCACF_ALIAS_Q_NAMES	3017	X'00000BC9'
MQCACF_REMOTE_Q_NAMES	3018	X'00000BCA'
MQCACF_SENDER_CHANNEL_NAMES	3019	X'00000BCB'
MQCACF_SERVER_CHANNEL_NAMES	3020	X'00000BCC'
MQCACF_REQUESTER_CHANNEL_NAMES	3021	X'00000BCD'
MQCACF_RECEIVER_CHANNEL_NAMES	3022	X'00000BCE'
MQCACF_OBJECT_Q_MGR_NAME	3023	X'00000BCF'
MQCACF_APPL_NAME	3024	X'00000BD0'
MQCACF_USER_IDENTIFIER	3025	X'00000BD1'
MQCACF_AUX_ERROR_DATA_STR_1	3026	X'00000BD2'
MQCACF_AUX_ERROR_DATA_STR_2	3027	X'00000BD3'
MQCACF_AUX_ERROR_DATA_STR_3	3028	X'00000BD4'
MQCACF_BRIDGE_NAME	3029	X'00000BD5'
MQCACF_STREAM_NAME	3030	X'00000BD6'
MQCACF_TOPIC	3031	X'00000BD7'
MQCACF_PARENT_Q_MGR_NAME	3032	X'00000BD8'
MQCACF_CORREL_ID	3033	X'00000BD9'
MQCACF_PUBLISH_TIMESTAMP	3034	X'00000BDA'
MQCACF_STRING_DATA	3035	X'00000BDB'
MQCACF_SUPPORTED_STREAM_NAME	3036	X'00000BDC'
MQCACF_REG_TOPIC	3037	X'00000BDD'
MQCACF_REG_TIME	3038	X'00000BDE'
MQCACF_REG_USER_ID	3039	X'00000BDF'
MQCACF_CHILD_Q_MGR_NAME	3040	X'00000BE0'
MQCACF_REG_STREAM_NAME	3041	X'00000BE1'
MQCACF_REG_Q_MGR_NAME	3042	X'00000BE2'
MQCACF_REG_Q_NAME	3043	X'00000BE3'
MQCACF_REG_CORREL_ID	3044	X'00000BE4'
MQCACF_EVENT_USER_ID	3045	X'00000BE5'
MQCACF_OBJECT_NAME	3046	X'00000BE6'
MQCACF_EVENT_Q_MGR	3047	X'00000BE7'
MQCACF_AUTH_INFO_NAMES	3048	X'00000BE8'
MQCACF_EVENT_APPL_IDENTITY	3049	X'00000BE9'
MQCACF_EVENT_APPL_NAME	3050	X'00000BEA'
MQCACF_EVENT_APPL_ORIGIN	3051	X'00000BEB'
MQCACF_SUBSCRIPTION_NAME	3052	X'00000BEC'

Tabela 40. Valores de constantes (continuação)

Nome	Valor decimal	Valor Hexadecimal
MQCACF_REG_SUB_NAME	3053	X'00000BED'
MQCACF_SUBSCRIPTION_IDENTITY	3054	X'00000BEE'
MQCACF_REG_SUB_IDENTITY	3055	X'00000BEF'
MQCACF_SUBSCRIPTION_USER_DATA	3056	X'00000BF0'
MQCACF_REG_SUB_USER_DATA	3057	X'00000BF1'
MQCACF_APPL_TAG	3058	X'00000BF2'
MQCACF_DATA_SET_NAME	3059	X'00000BF3'
MQCACF_UOW_START_DATE	3060	X'00000BF4'
MQCACF_UOW_START_TIME	3061	X'00000BF5'
MQCACF_UOW_LOG_START_DATE	3062	X'00000BF6'
MQCACF_UOW_LOG_START_TIME	3063	X'00000BF7'
MQCACF_UOW_LOG_EXTENT_NAME	3064	X'00000BF8'
MQCACF_PRINCIPAL_ENTITY_NAMES	3065	X'00000BF9'
MQCACF_GROUP_ENTITY_NAMES	3066	X'00000BFA'
MQCACF_AUTH_PROFILE_NAME	3067	X'00000BFB'
MQCACF_ENTITY_NAME	3068	X'00000BFC'
MQCACF_SERVICE_COMPONENT	3069	X'00000BFD'
MQCACF_RESPONSE_Q_MGR_NAME	3070	X'00000BFE'
MQCACF_CURRENT_LOG_EXTENT_NAME	3071	X'00000BFF'
MQCACF_RESTART_LOG_EXTENT_NAME	3072	X'00000C00'
MQCACF_MEDIA_LOG_EXTENT_NAME	3073	X'00000C01'
MQCACF_LOG_PATH	3074	X'00000C02'
MQCACF_COMMAND_MQSC	3075	X'00000C03'
MQCACF_Q_MGR_CPF	3076	X'00000C04'
MQCACF_USAGE_LOG_RBA	3078	X'00000C06'
MQCACF_USAGE_LOG_LRSN	3079	X'00000C07'
MQCACF_COMMAND_SCOPE	3080	X'00000C08'
MQCACF_ASID	3081	X'00000C09'
MQCACF_PSB_NAME	3082	X'00000C0A'
MQCACF_PST_ID	3083	X'00000C0B'
MQCACF_TASK_NUMBER	3084	X'00000C0C'
MQCACF_TRANSACTION_ID	3085	X'00000C0D'
MQCACF_Q_MGR_UOW_ID	3086	X'00000C0E'
MQCACF_ORIGIN_NAME	3088	X'00000C10'
MQCACF_ENV_INFO	3089	X'00000C11'
MQCACF_SECURITY_PROFILE	3090	X'00000C12'
MQCACF_CONFIGURATION_DATE	3091	X'00000C13'
MQCACF_CONFIGURATION_TIME	3092	X'00000C14'
MQCACF_FROM_CF_STRUC_NAME	3093	X'00000C15'

Tabela 40. Valores de constantes (continuação)

Nome	Valor decimal	Valor Hexadecimal
MQCACF_TO_CF_STRUC_NAME	3094	X'00000C16'
MQCACF_CF_STRUC_NAMES	3095	X'00000C17'
MQCACF_FAIL_DATE	3096	X'00000C18'
MQCACF_FAIL_TIME	3097	X'00000C19'
MQCACF_BACKUP_DATE	3098	X'00000C1A'
MQCACF_BACKUP_TIME	3099	X'00000C1B'
MQCACF_SYSTEM_NAME	3100	X'00000C1C'
MQCACF_CF_STRUC_BACKUP_START	3101	X'00000C1D'
MQCACF_CF_STRUC_BACKUP_END	3102	X'00000C1E'
MQCACF_CF_STRUC_LOG_Q_MGRS	3103	X'00000C1F'
MQCACF_FROM_STORAGE_CLASS	3104	X'00000C20'
MQCACF_TO_STORAGE_CLASS	3105	X'00000C21'
MQCACF_STORAGE_CLASS_NAMES	3106	X'00000C22'
MQCACF_DSG_NAME	3108	X'00000C24'
MQCACF_DB2_NAME	3109	X'00000C25'
MQCACF_SYSP_CMD_USER_ID	3110	X'00000C26'
MQCACF_SYSP_OTMA_GROUP	3111	X'00000C27'
MQCACF_SYSP_OTMA_MEMBER	3112	X'00000C28'
MQCACF_SYSP_OTMA_DRU_EXIT	3113	X'00000C29'
MQCACF_SYSP_OTMA_TPIPE_PFX	3114	X'00000C2A'
MQCACF_SYSP_ARCHIVE_PFX1	3115	X'00000C2B'
MQCACF_SYSP_ARCHIVE_UNIT1	3116	X'00000C2C'
MQCACF_SYSP_LOG_CORREL_ID	3117	X'00000C2D'
MQCACF_SYSP_UNIT_VOLSER	3118	X'00000C2E'
MQCACF_SYSP_Q_MGR_TIME	3119	X'00000C2F'
MQCACF_SYSP_Q_MGR_DATE	3120	X'00000C30'
MQCACF_SYSP_Q_MGR_RBA	3121	X'00000C31'
MQCACF_SYSP_LOG_RBA	3122	X'00000C32'
MQCACF_SYSP_SERVICE	3123	X'00000C33'
MQCACF_FROM_LISTENER_NAME	3124	X'00000C34'
MQCACF_TO_LISTENER_NAME	3125	X'00000C35'
MQCACF_FROM_SERVICE_NAME	3126	X'00000C36'
MQCACF_TO_SERVICE_NAME	3127	X'00000C37'
MQCACF_LAST_PUT_DATE	3128	X'00000C38'
MQCACF_LAST_PUT_TIME	3129	X'00000C39'
MQCACF_LAST_GET_DATE	3130	X'00000C3A'
MQCACF_LAST_GET_TIME	3131	X'00000C3B'
MQCACF_OPERATION_DATE	3132	X'00000C3C'
MQCACF_OPERATION_TIME	3133	X'00000C3D'

Tabela 40. Valores de constantes (continuação)

Nome	Valor decimal	Valor Hexadecimal
MQCACF_ACTIVITY_DESC	3134	X'00000C3E'
MQCACF_APPL_IDENTITY_DATA	3135	X'00000C3F'
MQCACF_APPL_ORIGIN_DATA	3136	X'00000C40'
MQCACF_PUT_DATE	3137	X'00000C41'
MQCACF_PUT_TIME	3138	X'00000C42'
MQCACF_REPLY_TO_Q	3139	X'00000C43'
MQCACF_REPLY_TO_Q_MGR	3140	X'00000C44'
MQCACF_RESOLVED_Q_NAME	3141	X'00000C45'
MQCACF_STRUC_ID	3142	X'00000C46'
MQCACF_VALUE_NAME	3143	X'00000C47'
MQCACF_SERVICE_START_DATE	3144	X'00000C48'
MQCACF_SERVICE_START_TIME	3145	X'00000C49'
MQCACF_SYSP_OFFLINE_RBA	3146	X'00000C4A'
MQCACF_SYSP_ARCHIVE_PFX2	3147	X'00000C4B'
MQCACF_SYSP_ARCHIVE_UNIT2	3148	X'00000C4C'
MQCACF_TO_TOPIC_NAME	3149	X'00000C4D'
MQCACF_FROM_TOPIC_NAME	3150	X'00000C4E'
MQCACF_TOPIC_NAMES	3151	X'00000C4F'
MQCACF_SUB_NAME	3152	X'00000C50'
MQCACF_DESTINATION_Q_MGR	3153	X'00000C51'
MQCACF_DESTINATION	3154	X'00000C52'
MQCACF_SUB_USER_ID	3156	X'00000C54'
MQCACF_SUB_USER_DATA	3159	X'00000C57'
MQCACF_SUB_SELECTOR	3160	X'00000C58'
MQCACF_LAST_PUB_DATE	3161	X'00000C59'
MQCACF_LAST_PUB_TIME	3162	X'00000C5A'
MQCACF_FROM_SUB_NAME	3163	X'00000C5B'
MQCACF_TO_SUB_NAME	3164	X'00000C5C'
MQCACF_LAST_MSG_TIME	3167	X'00000C5F'
MQCACF_LAST_MSG_DATE	3168	X'00000C60'
MQCACF_SUBSCRIPTION_POINT	3169	X'00000C61'
MQCACF_FILTER	3170	X'00000C62'
MQCACF_NONE	3171	X'00000C63'
MQCACF_ADMIN_TOPIC_NAMES	3172	X'00000C64'
MQCACF_ROUTING_FINGER_PRINT	3173	X'00000C65'
MQCACF_APPL_DESC	3174	X'00000C66'
MQCACF_Q_MGR_START_DATE	3175	X'00000C67'
MQCACF_Q_MGR_START_TIME	3176	X'00000C68'
MQCACF_FROM_COMM_INFO_NAME	3177	X'00000C69'

Tabela 40. Valores de constantes (continuação)

Nome	Valor decimal	Valor Hexadecimal
MQCACF_TO_COMM_INFO_NAME	3178	X'00000C6A'
MQCACF_CF_OFFLOAD_SIZE1	3179	X'00000C6B'
MQCACF_CF_OFFLOAD_SIZE2	3180	X'00000C6C'
MQCACF_CF_OFFLOAD_SIZE3	3181	X'00000C6D'
MQCACF_CF_SMDS_GENERIC_NAME	3182	X'00000C6E'
MQCACF_CF_SMDS	3183	X'00000C6F'
MQCACF_RECOVERY_DATE	3184	X'00000C70'
MQCACF_RECOVERY_TIME	3185	X'00000C71'
MQCACF_CF_SMDSCONN	3186	X'00000C72'
MQCACF_CF_STRUC_NAME	3187	X'00000C73'
MQCACF_ALTERNATE_USERID	3188	X'00000C74'
MQCACF_CHAR_ATTRS	3189	X'00000C75'
MQCACF_DYNAMIC_Q_NAME	3190	X'00000C76'
MQCACF_HOST_NAME	3191	X'00000C77'
MQCACF_MQCB_NAME	3192	X'00000C78'
MQCACF_OBJECT_STRING	3193	X'00000C79'
MQCACF_RESOLVED_LOCAL_Q_MGR	3194	X'00000C7A'
MQCACF_RESOLVED_LOCAL_Q_NAME	3195	X'00000C7B'
MQCACF_RESOLVED_OBJECT_STRING	3196	X'00000C7C'
MQCACF_RESOLVED_Q_MGR	3197	X'00000C7D'
MQCACF_SELECTION_STRING	3198	X'00000C7E'
MQCACF_XA_INFO	3199	X'00000C7F'
MQCACF_APPL_FUNCTION	3200	X'00000C80'
MQCACF_XQH_REMOTE_Q_NAME	3201	X'00000C81'
MQCACF_XQH_REMOTE_Q_MGR	3202	X'00000C82'
MQCACF_XQH_PUT_TIME	3203	X'00000C83'
MQCACF_XQH_PUT_DATE	3204	X'00000C84'
MQCACF_EXCL_OPERATOR_MESSAGES	3205	X'00000C85'
MQCACF_CSP_USER_IDENTIFIER	3206	X'00000C86'
MQCACF_AMQP_CLIENT_ID	3207	X'00000C87'
MQCACF_ARCHIVE_LOG_EXTENT_NAME	3208	X'00000C88'
MQCACF_APPL_IMMOVABLE_DATE	3209	X'00000C89'
MQCACF_APPL_IMMOVABLE_TIME	3210	X'00000C8A'
MQCACF_NHA_INSTANCE_NAME	3211	X'00000C8B'
MQCACF_LAST_USED	3211	X'00000C8B'

## MQCACH\_\* (Tipos de Parâmetro de Canal de Caractere de Formato de Comando)..

*Tabela 41. Valores de constantes*

Nome	Valor decimal	Valor Hexadecimal
MQCACH_FIRST	3501	X'0000DAD'
MQCACH_CHANNEL_NAME	3501	X'0000DAD'
MQCACH_DESC	3502	X'0000DAE'
MQCACH_MODE_NAME	3503	X'0000DAF'
MQCACH_TP_NAME	3504	X'0000DB0'
MQCACH_XMIT_Q_NAME	3505	X'0000DB1'
MQCACH_CONNECTION_NAME	3506	X'0000DB2'
MQCACH_MCA_NAME	3507	X'0000DB3'
MQCACH_SEC_EXIT_NAME	3508	X'0000DB4'
MQCACH_MSG_EXIT_NAME	3509	X'0000DB5'
MQCACH_SEND_EXIT_NAME	3510	X'0000DB6'
MQCACH_RCV_EXIT_NAME	3511	X'0000DB7'
MQCACH_CHANNEL_NAMES	3512	X'0000DB8'
MQCACH_SEC_EXIT_USER_DATA	3513	X'0000DB9'
MQCACH_MSG_EXIT_USER_DATA	3514	X'0000DBA'
MQCACH_SEND_EXIT_USER_DATA	3515	X'0000DBB'
MQCACH_RCV_EXIT_USER_DATA	3516	X'0000DBC'
MQCACH_USER_ID	3517	X'0000DBD'
MQCACH_PASSWORD	3518	X'0000DBE'
MQCACH_LOCAL_ADDRESS	3520	X'0000DC0'
MQCACH_LOCAL_NAME	3521	X'0000DC1'
MQCACH_LAST_MSG_TIME	3524	X'0000DC4'
MQCACH_LAST_MSG_DATE	3525	X'0000DC5'
MQCACH_MCA_USER_ID	3527	X'0000DC7'
MQCACH_CHANNEL_START_TIME	3528	X'0000DC8'
MQCACH_CHANNEL_START_DATE	3529	X'0000DC9'
MQCACH_MCA_JOB_NAME	3530	X'0000DCA'
MQCACH_LAST_LUWID	3531	X'0000DCB'
MQCACH_CURRENT_LUWID	3532	X'0000DCC'
MQCACH_FORMAT_NAME	3533	X'0000DCD'
MQCACH_MR_EXIT_NAME	3534	X'0000DCE'
MQCACH_MR_EXIT_USER_DATA	3535	X'0000DCF'
MQCACH_SSL_CIPHER_SPEC	3544	X'0000DD8'
MQCACH_SSL_PEER_NAME	3545	X'0000DD9'
MQCACH_SSL_HANDSHAKE_STAGE	3546	X'0000DDA'
MQCACH_SSL_SHORT_PEER_NAME	3547	X'0000ddb'
MQCACH_REMOTE_APPL_TAG	3548	X'0000DDC'

Tabela 41. Valores de constantes (continuação)

Nome	Valor decimal	Valor Hexadecimal
MQCACH_SSL_CERT_USER_ID	3549	X'00000DDD'
MQCACH_SSL_CERT_ISSUER_NAME	3550	X'00000DDE'
MQCACH_LU_NAME	3551	X'00000DDF'
MQCACH_IP_ADDRESS	3552	X'00000DE0'
MQCACH_TCP_NAME	3553	X'00000DE1'
MQCACH_LISTENER_NAME	3554	X'00000DE2'
MQCACH_LISTENER_DESC	3555	X'00000DE3'
MQCACH_LISTENER_START_DATE	3556	X'00000DE4'
MQCACH_LISTENER_START_TIME	3557	X'00000DE5'
MQCACH_SSL_KEY_RESET_DATE	3558	X'00000DE6'
MQCACH_SSL_KEY_RESET_TIME	3559	X'00000DE7'
MQCACH_LAST_USED	3559	X'00000DE7'

### **MQCADSD\_\* (Descritores ADS do cabeçalho de informações CICS)**

Tabela 42. Valores de constantes

Nome	Valor decimal	Valor Hexadecimal
MQCADSD_NONE	0	X'00000000'
MQCADSD_SEND	1	X'00000001'
MQCADSD_RECV	16	X'00000010'
MQCADSD_MSGFORMAT	256	X'00000100'

### **MQCAFTY\_\* (Valores de afinidade de conexão)**

Tabela 43. Valores de constantes

Nome	Valor decimal	Valor Hexadecimal
MQCAFTY_NONE	0	X'00000000'
MQCAFTY_PREFERRED	1	X'00000001'

### **MQCAMO\_\* (Tipos de Parâmetros de Monitoramento de Caracteres do Formato de Comando)**

Tabela 44. Valores de constantes

Nome	Valor decimal	Valor Hexadecimal
MQCAMO_FIRST	2701	X'00000A8D'
MQCAMO_CLOSE_DATE	2701	X'00000A8D'
MQCAMO_CLOSE_TIME	2702	X'00000A8E'
MQCAMO_CONN_DATE	2703	X'00000A8F'
MQCAMO_CONN_TIME	2704	X'00000A90'
MQCAMO_DISC_DATE	2705	X'00000A91'
MQCAMO_DISC_TIME	2706	X'00000A92'
MQCAMO_END_DATE	2707	X'00000A93'
MQCAMO_END_TIME	2708	X'00000A94'

<i>Tabela 44. Valores de constantes (continuação)</i>		
Nome	Valor decimal	Valor Hexadecimal
MQCAMO_OPEN_DATE	2709	X'00000A95'
MQCAMO_OPEN_TIME	2710	X'00000A96'
MQCAMO_START_DATE	2711	X'00000A97'
MQCAMO_START_TIME	2712	X'00000A98'
MQCAMO_LAST_USED	2712	X'00000A98'

### **MQCBC\_\* (estrutura de constantes MQCBC)**

<i>Tabela 45. Estruturas de constantes</i>	
Nome	Estrutura
MQCBC_STRUC_ID	"CBC~"
MQCBC_STRUC_ID_ARRAY	'C', 'B', 'C', '~'

**Nota:** O símbolo ~ representa um único caractere em branco.

<i>Tabela 46. Valores de constantes</i>		
Nome	Valor decimal	Valor Hexadecimal
MQCBC_VERSION_1	1	X'00000001'
MQCBC_CURRENT_VERSION	1	X'00000001'

### **MQCBCF\_\* (Sinalizadores de constantes MQCBC)**

<i>Tabela 47. Valores de constantes</i>		
Nome	Valor decimal	Valor Hexadecimal
MQCBCF_NONE	0	X'00000000'
MQCBCF_READA_BUFFER_EMPTY	1	X'00000001'

### **MQCBCT\_\* (tipo de retorno de chamada de constantes MQCBC)**

<i>Tabela 48. Valores de constantes</i>		
Nome	Valor decimal	Valor Hexadecimal
MQCBCT_START_CALL	1	X'00000001'
MQCBCT_STOP_CALL	2	X'00000002'
MQCBCT_REGISTER_CALL	3	X'00000003'
MQCBCT_DEREGISTER_CALL	4	X'00000004'
MQCBCT_EVENT_CALL	5	X'00000005'
MQCBCT_MSG_REMOVIDO	6	X'00000006'
MQCBCT_MSG_NOT_REMOVE	7	X'00000007'

### **MQCBD\_\* (estrutura de constantes MQCBD)**

<i>Tabela 49. Estruturas de constantes</i>	
Nome	Estrutura
MQCBD_STRUC_ID	"CBD~"
MQCBD_STRUC_ID_ARRAY	'C', 'B', 'D', '~'



**Nota:** O símbolo – representa um único caractere em branco.

<i>Tabela 50. Valores de constantes</i>		
Nome	Valor decimal	Valor Hexadecimal
MQCBDD_VERSION_1	1	X'00000001'
MQCBDD_CURRENT_VERSION	1	X'00000001'

### **MQCBDO\_\* (Opções de Retorno de Chamada de constantes MQCBDD)**

<i>Tabela 51. Valores de constantes</i>		
Nome	Valor decimal	Valor Hexadecimal
MQCBDO_NONE	0	X'00000000'
MQCBDO_START_CALL	1	X'00000001'
MQCBDO_STOP_CALL	4	X'00000004'
MQCBDO_REGISTER_CALL	256	X'00000100'
MQCBDO_DEREGISTER_CALL	512	X'00000200'
MQCBDO_FAIL_IF QUIESCING	8192	X'00002000'

### **MQCBO\_\* (Opções Create-Bag para mqCreateBag)**

<i>Tabela 52. Valores de constantes</i>		
Nome	Valor decimal	Valor Hexadecimal
MQCBO_NONE	0	X'00000000'
MQCBO_USER_BAG	0	X'00000000'
MQCBO_ADMIN_BAG	1	X'00000001'
MQCBO_COMMAND_BAG	16	X'00000010'
MQCBO_SYSTEM_BAG	32	X'00000020'
MQCBO_GROUP_BAG	64	X'00000040'
MQCBO_LIST_FORM_ALLOWED	2	X'00000002'
MQCBO_LIST_FORM_INIBIDA	0	X'00000000'
MQCBO_REORDER_AS_REQUIRED	4	X'00000004'
MQCBO_DO_NOT_REORDER	0	X'00000000'
MQCBO_CHECK_SELECTORS	8	X'00000008'
MQCBO_DO_NOT_CHECK_SELECTORS	0	X'00000000'

### **MQCBT\_\* (constantes MQCBDD Este é o tipo da Função de Retorno de Chamada)**

<i>Tabela 53. Valores de constantes</i>		
Nome	Valor decimal	Valor Hexadecimal
MQCBT_MESSAGE_CONSUMER	1	X'00000001'
MQCBT_EVENT_HANDLER	2	X'00000002'

## MQCC\_\* (códigos de conclusão)

*Tabela 54. Valores de constantes*

Nome	Valor decimal	Valor Hexadecimal
MQCC_OK	0	X'00000000'
MQCC_WARNING	1	X'00000001'
MQCC_FAILED	2	X'00000002'
MQCC_UNKNOWN	-1	X'FFFFFFFF'

## MQCCSI\_\* (Identificadores do conjunto de caracteres codificados)

*Tabela 55. Valores de constantes*

Nome	Valor decimal	Valor Hexadecimal
MQCCSI_UNDEFINED	0	X'00000000'
MQCCSI_DEFAULT	0	X'00000000'
MQCCSI_Q_MGR	0	X'00000000'
MQCCSI_INHERIT	-2	X'FFFFFFFE'
MQCCSI_EMBEDDED	-1	X'FFFFFFFF'
MQCCSI_APPL	-3	X'FFFFFFFD'




## MQCCT\_\* (CICS Informações do cabeçalho Conversational Task Options)

*Tabela 56. Valores de constantes*



Nome	Valor decimal	Valor Hexadecimal
MQCCT_YES	1	X'00000001'
MQCCT_NO	0	X'00000000'

## MQCD\_\* (Estrutura de definição de canal).

*Tabela 57. Valores de constantes*

Nome	Valor decimal	Valor Hexadecimal
MQCD_VERSION_1	1	X'00000001'
MQCD_VERSION_2	2	X'00000002'
MQCD_VERSION_3	3	X'00000003'
MQCD_VERSION_4	4	X'00000004'
MQCD_VERSION_5	5	X'00000005'
MQCD_VERSION_6	6	X'00000006'
MQCD_VERSION_7	7	X'00000007'
MQCD_VERSION_8	8	X'00000008'
MQCD_VERSION_9	9	X'00000009'
MQCD_VERSION_10	10	X'0000000A'
 MQCD_VERSION_11	11	X'0000000B'
 MQCD_CURRENT_VERSION	11	X'0000000B'
 MQCD_VERSION_12	12	X'0000000C'

*Tabela 57. Valores de constantes (continuação)*

Nome	Valor decimal	Valor Hexadecimal
 MQCD_CURRENT_VERSION	12	X'0000000C'
MQCD_LENGTH_4	(value differs by platform or version)	(value differs by platform or version)
MQCD_LENGTH_5	(value differs by platform or version)	(value differs by platform or version)
MQCD_LENGTH_6	(value differs by platform or version)	(value differs by platform or version)
MQCD_LENGTH_7	(value differs by platform or version)	(value differs by platform or version)
MQCD_LENGTH_8	(value differs by platform or version)	(value differs by platform or version)
MQCD_LENGTH_9	(value differs by platform or version)	(value differs by platform or version)
MQCD_LENGTH_10	(value differs by platform or version)	(value differs by platform or version)
MQCD_LENGTH_11	(value differs by platform or version)	(value differs by platform or version)
 MQCD_LENGTH_12	(value differs by platform or version)	(value differs by platform or version)
MQCD_CURRENT_LENGTH	(value differs by platform or version)	(value differs by platform or version)

### MQCDC\_\* (Conversão de Dados do Canal).

*Tabela 58. Valores de constantes*

Nome	Valor decimal	Valor Hexadecimal
MQCDC_SENDER_CONVERSION	1	X'00000001'
MQCDC_NO_SENDER_CONVERSION	0	X'00000000'

### MQCERT\_\* (Tipo de política de validação de certificado)

MQ_CERT_VAL_POLICY_DEFAULT	0	X'00000000'
MQ_CERT_VAL_POLICY_ANY	0	X'00000000'
MQ_CERT_VAL_POLICY_RFC5280	1	X'00000001'

### MQCF\_\* (Sinalizações de Recurso)

*Tabela 59. Valores de constantes*

Nome	Valor decimal	Valor Hexadecimal
MQCF_NONE	0	X'00000000'
MQCF_DIST_LISTS	1	X'00000001'

### MQCFAC\_\* (Recurso de cabeçalho de informações do CICS)

*Tabela 60. Nomes e valores de constantes*

Nome	Valor Hexadecimal
MQCFAC_NONE	X'00...00' (8 nulos)

Tabela 60. Nomes e valores de constantes (continuação)	
Nome	Valor Hexadecimal
MQCFAC_NONE_ARRAY	'\0', '\0', ... (8 nulos)

### **MQCFBF\_\* (Estrutura de parâmetro de filtro de sequência de bytes de formato de comando).**

Tabela 61. Valores de constantes		
Nome	Valor decimal	Valor Hexadecimal
MQCFBF_STRUC_LENGTH_FIXED	20	X'00000014'

### **MQCFBS\_\* (Estrutura do parâmetro de sequência de bytes do formato de comando)..**

Tabela 62. Valores de constantes		
Nome	Valor decimal	Valor Hexadecimal
MQCFBS_STRUC_LENGTH_FIXED	16	X'00000010'

### **MQCF\*\_\* (Opções de controle do cabeçalho do formato de comando)**

Tabela 63. Valores de constantes		
Nome	Valor decimal	Valor Hexadecimal
MQCF*_LAST	1	X'00000001'
MQCF*_NOT_LAST	0	X'00000000'

### **MQCFGR\_\* (Estrutura de parâmetro do grupo de formato de comando).**

Tabela 64. Valores de constantes		
Nome	Valor decimal	Valor Hexadecimal
MQCFGR_STRUC_LENGTH	16	X'00000010'

### **MQCFH\_\* (Estrutura do cabeçalho do formato de comando)**

Tabela 65. Valores de constantes		
Nome	Valor decimal	Valor Hexadecimal
MQCFH_STRUC_LENGTH	36	X'00000024'
MQCFH_VERSION_1	1	X'00000001'
MQCFH_VERSION_2	2	X'00000002'
MQCFH_VERSION_3	3	X'00000003'
MQCFH_CURRENT_VERSION	3	X'00000003'

### **MQCFIF\_\* (Estrutura do parâmetro de filtro de número inteiro do formato de comando)..**

Tabela 66. Valores de constantes		
Nome	Valor decimal	Valor Hexadecimal
MQCFIF_STRUC_LENGTH	20	X'00000014'

## **MQCFIL\_\* (Estrutura de parâmetro da lista de inteiros do formato de comando)**

<i>Tabela 67. Valores de constantes</i>		
<b>Nome</b>	<b>Valor decimal</b>	<b>Valor Hexadecimal</b>
MQCFIL_STRUC_LENGTH_FIXED	16	X'00000010'

## **MQCFIL64\_\* (Estrutura de parâmetro de lista de números inteiros de 64 bits no formato de comando)**

<i>Tabela 68. Valores de constantes</i>		
<b>Nome</b>	<b>Valor decimal</b>	<b>Valor Hexadecimal</b>
MQCFIL64_STRUC_LENGTH_FIXED	16	X'00000010'

## **MQCFIN\_\* (Estrutura de parâmetro de número inteiro do formato de comando).**

<i>Tabela 69. Valores de constantes</i>		
<b>Nome</b>	<b>Valor decimal</b>	<b>Valor Hexadecimal</b>
MQCFIN_STRUC_LENGTH	16	X'00000010'

## **MQCFIN64\_\* (Estrutura de parâmetro de número inteiro do formato de comando de 64 bits).**

<i>Tabela 70. Valores de constantes</i>		
<b>Nome</b>	<b>Valor decimal</b>	<b>Valor Hexadecimal</b>
MQCFIN64_STRUC_LENGTH	24	X'00000018'

## **MQCFO\_\* (Formato de Comando Opções de Repositório de Atualização e Formato de Comando Remove Opções de Filas)**

### **Opções do Repositório de Atualização do formato de comando**

<i>Tabela 71. Valores de constantes</i>		
<b>Nome</b>	<b>Valor decimal</b>	<b>Valor Hexadecimal</b>
MQCFO_REFRESH_REPOSITORY_YES	1	X'00000001'
MQCFO_REFRESH_REPOSITORY_NO	0	X'00000000'

### **Opções de Remoção de Filas do Formato de Comando..**

<i>Tabela 72. Valores de constantes</i>		
<b>Nome</b>	<b>Valor decimal</b>	<b>Valor Hexadecimal</b>
MQCFO_REMOVE_QUEUES_YES	1	X'00000001'
MQCFO_REMOVE_QUEUES_NO	0	X'00000000'

## MQCFOP\_\* (Operadores de Filtro de formato de comando)

Tabela 73. Valores de constantes		
Nome	Valor decimal	Valor Hexadecimal
MQCFOP_LESS	1	X'00000001'
MQCFOP_EQUAL	2	X'00000002'
MQCFOP_MAIOR	4	X'00000004'
MQCFOP_NOT_LESS	6	X'00000006'
MQCFOP_NOT_EQUAL	5	X'00000005'
MQCFOP_NOT_MAIOR	3	X'00000003'
MQCFOP_LIKE	18	X'00000012'
MQCFOP_NOT_LIKE	21	X'00000015'
MQCFOP_CONTAINS	10	X'0000000A'
MQCFOP_EXINCLUI	13	X'0000000D'
MQCFOP_CONTAINS_GEN	26	X'0000001A'
MQCFOP_EXCLUDES_GEN	29	X'0000001D'

## MQCFR\_\* (Recuperação de CF)

Tabela 74. Valores de constantes		
Nome	Valor decimal	Valor Hexadecimal
MQCFR_YES	1	X'00000001'
MQCFR_NO	0	X'00000000'

## MQCFSF\_\* (Estrutura do parâmetro de filtro de sequência de formato de comando)

Tabela 75. Valores de constantes		
Nome	Valor decimal	Valor Hexadecimal
MQCFSF_STRUC_LENGTH_FIXED	24	X'00000018'

## MQCFSL\_\* (Estrutura do parâmetro da lista de sequências de formato de comando)

Tabela 76. Valores de constantes		
Nome	Valor decimal	Valor Hexadecimal
MQCFSL_STRUC_LENGTH_FIXED	24	X'00000018'

## MQCFST\_\* (Estrutura do parâmetro de sequência de formato de comando)

Tabela 77. Valores de constantes		
Nome	Valor decimal	Valor Hexadecimal
MQCFST_STRUC_LENGTH_FIXED	20	X'00000014'

## MQCFSTATUS\_\* (Status do CF do formato de comando).

*Tabela 78. Valores de constantes*

Nome	Valor decimal	Valor Hexadecimal
MQCFSTATUS_NOT_FOUND	0	X'00000000'
MQCFSTATUS_ACTIVE	1	X'00000001'
MQCFSTATUS_IN_RECOVER	2	X'00000002'
MQCFSTATUS_IN_BACKUP	3	X'00000003'
MQCFSTATUS_FAILED	4	X'00000004'
MQCFSTATUS_NONE	5	X'00000005'
MQCFSTATUS_UNKNOWN	6	X'00000006'
MQCFSTATUS_ADMIN_INCOMPLETO	20	X'00000014'
MQCFSTATUS_NEVER_USED	21	X'00000015'
MQCFSTATUS_NO_BACKUP	22	X'00000016'
MQCFSTATUS_NOT_FAILED	23	X'00000017'
MQCFSTATUS_NOT_RECOVERABLE	24	X'00000018'
MQCFSTATUS_XES_ERROR	25	X'00000019'

## MQCFT\_\* (Tipos de Estrutura de Formato de Comandos).

*Tabela 79. Valores de constantes*

Nome	Valor decimal	Valor Hexadecimal
MQCFT_NONE	0	X'00000000'
MQCFT_COMMAND	1	X'00000001'
MQCFT_RESPONSE	2	X'00000002'
MQCFT_INTEGER	3	X'00000003'
MQCFT_STRING	4	X'00000004'
MQCFT_INTEGER_LIST	5	X'00000005'
MQCFT_STRING_LIST	6	X'00000006'
MQCFT_EVENT	7	X'00000007'
MQCFT_USER	8	X'00000008'
MQCFT_BYTE_STRING	9	X'00000009'
MQCFT_TRACE_ROUTE	10	X'0000000A'
MQCFT_REPORT	12	X'0000000C'
MQCFT_INTEGER_FILTER	13	X'0000000D'
MQCFT_STRING_FILTER	14	X'0000000E'
MQCFT_BYTE_STRING_FILTER	15	X'0000000F'
MQCFT_COMMAND_XR	16	X'00000010'
MQCFT_XR_MSG	17	X'00000011'
MQCFT_XR_ITEM	18	X'00000012'
MQCFT_XR_SUMMARY	19	X'00000013'
MQCFT_GROUP	20	X'00000014'
MQCFT_STATISTICS	21	X'00000015'

Tabela 79. Valores de constantes (continuação)		
Nome	Valor decimal	Valor Hexadecimal
MQCFT_ACCOUNTING	22	X'00000016'
MQCFT_INTEGER64	23	X'00000017'
MQCFT_INTEGER64_LIST	25	X'00000019'

### **MQCFTYPE\_\* (Tipos de CF do formato de comando).**

Tabela 80. Valores de constantes		
Nome	Valor decimal	Valor Hexadecimal
MQCFTYPE_APPL	0	X'00000000'
MQCFTYPE_ADMIN	1	X'00000001'

### **MQCFUNC\_\* (Funções do cabeçalho de informação CICS)**

Tabela 81. Estruturas de constantes	
Nome	Estrutura
MQCFUNC_MQCONN	"CONN"
MQCFUNC_MQGET	"GET~"
MQCFUNC_MQINQ	"INQ~"
MQCFUNC_MQOPEN	"OPEN"
MQCFUNC_MQPUT	"PUT~"
MQCFUNC_MQPUT1	"PUT1"
MQCFUNC_NONE	"~ ~ ~ ~"
MQCFUNC_MQCONN_ARRAY	'C','O','N','N'
MQCFUNC_MQGET_ARRAY	'G','E','T','~'
MQCFUNC_MQINQ_ARRAY	'I','N','Q','~'
MQCFUNC_MQOPEN_ARRAY	'O','P','E','N'
MQCFUNC_MQPUT_ARRAY	'P','U','T','~'
MQCFUNC_MQPUT1_ARRAY	'P','U','T','1'
MQCFUNC_NONE_ARRAY	'~','~','~','~'

**Nota:** O símbolo ~ representa um único caractere em branco.

### **MQCGWI\_\* (Intervalo de Espera de Obtenção do cabeçalho de informações CICS)**

Tabela 82. Valores de constantes		
Nome	Valor decimal	Valor Hexadecimal
MQCGWI_DEFAULT	-2	X'FFFFFFFE'

### **MQCHAD\_\* (Definição automática de canal)**

Tabela 83. Valores de constantes		
Nome	Valor decimal	Valor Hexadecimal
MQCHAD_DISABLED	0	X'00000000'



<i>Tabela 83. Valores de constantes (continuação)</i>		
<b>Nome</b>	<b>Valor decimal</b>	<b>Valor Hexadecimal</b>
MQCHAD_ENABLED	1	X'00000001'

### **MQCHIDS\_\* (Status Indeterminado do Formato de Comando).**

<i>Tabela 84. Valores de constantes</i>		
<b>Nome</b>	<b>Valor decimal</b>	<b>Valor Hexadecimal</b>
MQCHIDS_NOT_INDOUBT	0	X'00000000'
MQCHIDS_INDOUBT	1	X'00000001'

### **MQCHLD\_\* (Disposições do Canal de Formato de Comandos)**

<i>Tabela 85. Valores de constantes</i>		
<b>Nome</b>	<b>Valor decimal</b>	<b>Valor Hexadecimal</b>
MQCHLD_ALL	-1	X'FFFFFFFF'
MQCHLD_DEFAULT	1	X'00000001'
MQCHLD_SHARED	2	X'00000002'
MQCHLD_PRIVATE	4	X'00000004'
MQCHLD_FIXSHARED	5	X'00000005'

### **MQCHS\_\* (Status do Canal do formato de comando)**

<i>Tabela 86. Valores de constantes</i>		
<b>Nome</b>	<b>Valor decimal</b>	<b>Valor Hexadecimal</b>
MQCHS_INACTIVE	0	X'00000000'
MQCHS_BINDING	1	X'00000001'
MQCHS_STARTING	2	X'00000002'
MQCHS_RUNNING	3	X'00000003'
MQCHS_STOPPING	4	X'00000004'
MQCHS_RETRYING	5	X'00000005'
MQCHS_STOPPED	6	X'00000006'
MQCHS_REQUESTING	7	X'00000007'
MQCHS_PAUSED	8	X'00000008'
MQCHS_INITIALIZING	13	X'0000000D'
MQCHS_SWITCHING	14	X'0000000E'

### **MQCHSH\_\* (Opções de Reinicialização Compartilhada do Canal no formato de Comando)..**

<i>Tabela 87. Valores de constantes</i>		
<b>Nome</b>	<b>Valor decimal</b>	<b>Valor Hexadecimal</b>
MQCHSH_RESTART_NO	0	X'00000000'
MQCHSH_RESTART_YES	1	X'00000001'

## **MQCHSR\_\* (Opções de Parada do Canal do formato de comando).**

*Tabela 88. Valores de constantes*

<b>Nome</b>	<b>Valor decimal</b>	<b>Valor Hexadecimal</b>
MQCHSR_STOP_NOT_REQUESTED	0	X'00000000'
MQCHSR_STOP_REQUESTED	1	X'00000001'

## **MQCHSSTATE\_\* (Subestados do Canal do formato de comando)..**

*Tabela 89. Valores de constantes*

<b>Nome</b>	<b>Valor decimal</b>	<b>Valor Hexadecimal</b>
MQCHSSTATE_OTHER	0	X'00000000'
MQCHSSTATE_END_OF_BATCH	100	X'00000064'
MQCHSSTATE_ENVIANDO	200	X'000000C8'
MQCHSSTATE_RECEBIMENTO	300	X'0000012C'
MQCHSSTATE_SERIALIZING	400	X'00000190'
MQCHSSTATE_RESYNCHING	500	X'000001F4'
MQCHSSTATE_PULSAÇÃO	600	X'00000258'
MQCHSSTATE_IN_SCYEXIT	700	X'000002BC'
MQCHSSTATE_IN_RCVEXIT	800	X'00000320'
MQCHSSTATE_IN_SENDEXIT	900	X'00000384'
MQCHSSTATE_IN_MSGEXIT	1000	X'000003E8'
MQCHSSTATE_IN_MREXIT	1100	X'0000044C'
MQCHSSTATE_IN_CHADEXIT	1200	X'000004B0'
MQCHSSTATE_NET_CONECTAR	1250	X'000004E2'
MQCHSSTATE_SSL_HANDSHAKE	1300	X'00000514'
MQCHSSTATE_NAME_SERVER	1400	X'00000578'
MQCHSSTATE_IN_MQPUT	1500	X'000005DC'
MQCHSSTATE_IN_MQGET	1600	X'00000640'
MQCHSSTATE_IN_MQI_CALL	1700	X'000006A4'
MQCHSSTATE_COMPRESSIONANDO	1800	X'00000708'

## **MQCHT\_\* (Tipos de Canal)**

*Tabela 90. Valores de constantes*

<b>Nome</b>	<b>Valor decimal</b>	<b>Valor Hexadecimal</b>
MQCHT_SENDER	1	X'00000001'
MQCHT_SERVER	2	X'00000002'
MQCHT_RECEIVER	3	X'00000003'
MQCHT_REQUESTER	4	X'00000004'
MQCHT_ALL	5	X'00000005'
MQCHT_CLNTCONN	6	X'00000006'
MQCHT_SVRCONN	7	X'00000007'
MQCHT_CLUSRCVR	8	X'00000008'

Tabela 90. Valores de constantes (continuação)		
Nome	Valor decimal	Valor Hexadecimal
MQCHT_CLUSSDR	9	X'00000009'

### **MQHTAB\_\* (Tipos de Tabela de Canal de Formato de Comandos)**

Tabela 91. Valores de constantes		
Nome	Valor decimal	Valor Hexadecimal
MQHTAB_Q_MGR	1	X'00000001'
MQHTAB_CLNTCONN	2	X'00000002'

### **MQCI\_\* (Identificador de Correlação).**

Tabela 92. Nomes e valores de constantes	
Nome	Value
MQCI_NONE	X'00...00' (24 nulos)
MQCI_NONE_ARRAY	'\0', '\0', ... (24 nulos)
MQCI_NEW_SESSION	X'414D5121...'
MQCI_NEW_SESSION_ARRAY	'\x41', '\x4D', '\51', '\x21', ...

### **MQCIH\_\* (estrutura do cabeçalho de informações CICS e Sinalizadores)**

#### **Estrutura do cabeçalho de informações CICS**

Tabela 93. Estruturas de constantes	
Nome	Estrutura
MQCIH_STRUC_ID	"CIH~"
MQCIH_STRUC_ID_ARRAY	'C', 'I', 'H', '~'

**Nota:** O símbolo ~ representa um único caractere em branco.

Tabela 94. Valores de constantes		
Nome	Valor decimal	Valor Hexadecimal
MQCIH_VERSION_1	1	X'00000001'
MQCIH_VERSION_2	2	X'00000002'
MQCIH_CURRENT_VERSION	2	X'00000002'
MQCIH_LENGTH_1	164	X'000000A4'
MQCIH_LENGTH_2	180	X'000000B4'
MQCIH_CURRENT_LENGTH	180	X'000000B4'

#### **Sinalizadores do cabeçalho de informações CICS**

Tabela 95. Valores de constantes		
Nome	Valor decimal	Valor Hexadecimal
MQCIH_NONE	0	X'00000000'
MQCIH_PASS_EXPIRATION	1	X'00000001'
MQCIH_UNLIMITED_EXPIRATION	0	X'00000000'

<i>Tabela 95. Valores de constantes (continuação)</i>		
Nome	Valor decimal	Valor Hexadecimal
MQCIH_REPLY_WITHOUT_NULLS	2	X'00000002'
MQCIH_REPLY_WITH_NULLS	0	X'00000000'
MQCIH_SYNC_ON_RETURN	4	X'00000004'
MQCIH_NO_SYNC_ON_RETURN	0	X'00000000'

### **MQCLCT\_\* (Tipos de Cache de Cluster).**

<i>Tabela 96. Valores de constantes</i>		
Nome	Valor decimal	Valor Hexadecimal
MQCLCT_STATIC	0	X'00000000'
MQCLCT_DYNAMIC	1	X'00000001'

### **MQCLRS\_\* (Formato de Comando Limpar Escopo da Sequência de Tópicos)**

<i>Tabela 97. Valores de constantes</i>		
Nome	Valor decimal	Valor Hexadecimal
MQCLRS_LOCAL	1	X'00000001'
MQCLRS_GLOBAL	2	X'00000002'

### **MQCLRT\_\* (Formato de Comando Limpar Tipo de Sequência de Tópicos)**

<i>Tabela 98. Valores de constantes</i>		
Nome	Valor decimal	Valor Hexadecimal
MQCLRT_RETIDA	1	X'00000001'

### **MQCLT\_\* ( CICS tipos de link de cabeçalho de informações)**

<i>Tabela 99. Valores de constantes</i>		
Nome	Valor decimal	Valor Hexadecimal
MQCLT_PROGRAM	1	X'00000001'
MQCLT_TRANSACTION	2	X'00000002'

### **MQCLWL\_\* (Carga de Trabalho do Cluster).**

<i>Tabela 100. Valores de constantes</i>		
Nome	Valor decimal	Valor Hexadecimal
MQCLWL_USEQ_LOCAL	0	X'00000000'
MQCLWL_USEQ_ANY	1	X'00000001'
MQCLWL_USEQ_AS_Q_MGR	-3	X'FFFFFFFD'

### **MQCLXQ\_\* (Tipo de fila de transmissão do cluster).**

MQCLXQ\_\* são os valores que podem ser configurados no atributo do gerenciador de filas DEFCLXQ O atributo **DEFCLXQ** controla qual fila de transmissão é selecionada por padrão pelos canais do emissor de clusters para obter mensagens, para enviar as mensagens para canais do receptor de clusters.

Tabela 101. Valores de constantes		
Nome	Valor decimal	Valor Hexadecimal
MQCLXQ_SCTQ	0	X'00000000'
MQCLXQ_CHANNEL	1	X'00000001'

### Referências relacionadas

“DefClusterXmitQueueTipo (MQLONG)” na página 845

O atributo DefClusterXmitQueueTipo controla qual fila de transmissão é selecionada por padrão pelos canais do emissor de clusters para obter mensagens, para enviar as mensagens para os canais do receptor de clusters.

[Alterar Gerenciador de Filas](#)

[Consultar Gerenciador de Filas](#)

[Consultar Gerenciador de Filas \(Resposta\)](#)

“MQINQ-Consultar atributos do objeto.” na página 730

A chamada MQINQ retorna uma matriz de números inteiros e um conjunto de cadeias de caracteres que contêm os atributos de um objeto

### MQCMD\_\* (Códigos de Comando).

Tabela 102. Valores de constantes		
Nome	Valor decimal	Valor Hexadecimal
MQCMD_NONE	0	X'00000000'
MQCMD_CHANGE_Q_MGR	1	X'00000001'
MQCMD_INQUIRE_Q_MGR	2	X'00000002'
MQCMD_CHANGE_PROCESS	3	X'00000003'
MQCMD_COPY_PROCESS	4	X'00000004'
MQCMD_CREATE_PROCESS	5	X'00000005'
MQCMD_DELETE_PROCESS	6	X'00000006'
MQCMD_INQUIRE_PROCESS	7	X'00000007'
MQCMD_CHANGE_Q	8	X'00000008'
MQCMD_CLEAR_Q	9	X'00000009'
MQCMD_COPY_Q	10	X'0000000A'
MQCMD_CREATE_Q	11	X'0000000B'
MQCMD_DELETE_Q	12	X'0000000C'
MQCMD_INQUIRE_Q	13	X'0000000D'
MQCMD_REFRESH_Q_MGR	16	X'00000010'
MQCMD_RESET_Q_STATS	17	X'00000011'
MQCMD_INQUIRE_Q_NAMES	18	X'00000012'
MQCMD_INQUIRE_PROCESS_NAMES	19	X'00000013'
MQCMD_INQUIRE_CHANNEL_NAMES	20	X'00000014'
MQCMD_CHANGE_CHANNEL	21	X'00000015'
MQCMD_COPY_CHANNEL	22	X'00000016'
MQCMD_CREATE_CHANNEL	23	X'00000017'
MQCMD_DELETE_CHANNEL	24	X'00000018'
MQCMD_INQUIRE_CHANNEL	25	X'00000019'

Tabela 102. Valores de constantes (continuação)

Nome	Valor decimal	Valor Hexadecimal
MQCMD_PING_CHANNEL	26	X'0000001A'
MQCMD_RESET_CHANNEL	27	X'0000001B'
MQCMD_START_CHANNEL	28	X'0000001C'
MQCMD_STOP_CHANNEL	29	X'0000001D'
MQCMD_START_CHANNEL_INIT	30	X'0000001E'
MQCMD_START_CHANNEL_LISTENER	31	X'0000001F'
MQCMD_CHANGE_NAMELIST	32	X'00000020'
MQCMD_COPY_NAMELIST	33	X'00000021'
MQCMD_CREATE_NAMELIST	34	X'00000022'
MQCMD_DELETE_NAMELIST	35	X'00000023'
MQCMD_INQUIRE_NAMELIST	36	X'00000024'
MQCMD_INQUIRE_NAMELIST_NAMES	37	X'00000025'
MQCMD_ESCAPE	38	X'00000026'
MQCMD_RESOLVE_CHANNEL	39	X'00000027'
MQCMD_PING_Q_MGR	40	X'00000028'
MQCMD_INQUIRE_Q_STATUS	41	X'00000029'
MQCMD_INQUIRE_CHANNEL_STATUS	42	X'0000002A'
MQCMD_CONFIG_EVENT	43	X'0000002B'
MQCMD_Q_MGR_EVENT	44	X'0000002C'
MQCMD_PERFM_EVENT	45	X'0000002D'
MQCMD_CHANNEL_EVENT	46	X'0000002E'
MQCMD_DELETE_PUBLICATION	60	X'0000003C'
MQCMD_DEREGISTER_PUBLISHER	61	X'0000003D'
MQCMD_DEREGISTER_SUBSCRIBER	62	X'0000003E'
MQCMD_PUBLISH	63	X'0000003F'
MQCMD_REGISTER_PUBLISHER	64	X'00000040'
MQCMD_REGISTER_ASSINANTE	65	X'00000041'
MQCMD_REQUEST_UPDATE	66	X'00000042'
MQCMD_BROKER_INTERNAL	67	X'00000043'
MQCMD_ACTIVITY_MSG	69	X'00000045'
MQCMD_INQUIRE_CLUSTER_Q_MGR	70	X'00000046'
MQCMD_RESUME_Q_MGR_CLUSTER	71	X'00000047'
MQCMD_SUSPEND_Q_MGR_CLUSTER	72	X'00000048'
MQCMD_REFRESH_CLUSTER	73	X'00000049'
MQCMD_RESET_CLUSTER	74	X'0000004A'
MQCMD_TRACE_ROUTE	75	X'0000004B'
MQCMD_REFRESH_SECURITY	78	X'0000004E'
MQCMD_CHANGE_AUTH_INFO	79	X'0000004F'
MQCMD_COPY_AUTH_INFO	80	X'00000050'

Tabela 102. Valores de constantes (continuação)

Nome	Valor decimal	Valor Hexadecimal
MQCMD_CREATE_AUTH_INFO	81	X'00000051'
MQCMD_DELETE_AUTH_INFO	82	X'00000052'
MQCMD_INQUIRE_AUTH_INFO	83	X'00000053'
MQCMD_INQUIRE_AUTH_INFO_NAMES	84	X'00000054'
MQCMD_INQUIRE_CONNECTION	85	X'00000055'
MQCMD_STOP_CONNECTION	86	X'00000056'
MQCMD_INQUIRE_AUTH_RECS	87	X'00000057'
MQCMD_INQUIRE_ENTITY_AUTH	88	X'00000058'
MQCMD_DELETE_AUTH_REC	89	X'00000059'
MQCMD_SET_AUTH_REC	90	X'0000005A'
MQCMD_LOGGER_EVENT	91	X'0000005B'
MQCMD_RESET_Q_MGR	92	X'0000005C'
MQCMD_CHANGE_LISTENER	93	X'0000005D'
MQCMD_COPY_LISTENER	94	X'0000005E'
MQCMD_CREATE_LISTENER	95	X'0000005F'
MQCMD_DELETE_LISTENER	96	X'00000060'
MQCMD_INQUIRE_LISTENER	97	X'00000061'
MQCMD_INQUIRE_LISTENER_STATUS	98	X'00000062'
MQCMD_COMMAND_EVENT	99	X'00000063'
MQCMD_CHANGE_SECURITY	100	X'00000064'
MQCMD_CHANGE_CF_ESTRUTURA	101	X'00000065'
MQCMD_CHANGE_STG_CLASS	102	X'00000066'
MQCMD_CHANGE_TRACE	103	X'00000067'
MQCMD_ARCHIVE_LOG	104	X'00000068'
MQCMD_BACKUP_CF_ESTRUTURA	105	X'00000069'
MQCMD_CREATE_BUFFER_POOL	106	X'0000006A'
MQCMD_CREATE_PAGE_SET	107	X'0000006B'
MQCMD_CREATE_CF_ESTRUTURA	108	X'0000006C'
MQCMD_CREATE_STG_CLASS	109	X'0000006D'
MQCMD_COPY_CF_STRUC	110	X'0000006E'
MQCMD_COPY_STG_CLASS	111	X'0000006F'
MQCMD_DELETE_CF_STRUC	112	X'00000070'
MQCMD_DELETE_STG_CLASS	113	X'00000071'
MQCMD_INQUIRE_ARCHIVE	114	X'00000072'
MQCMD_INQUIRE_CF_ESTRUTURA	115	X'00000073'
MQCMD_INQUIRE_CF_STRUC_STATUS	116	X'00000074'
MQCMD_INQUIRE_CMD_SERVER	117	X'00000075'
MQCMD_INQUIRE_CHANNEL_INIT	118	X'00000076'
MQCMD_INQUIRE_QSG	119	X'00000077'

Tabela 102. Valores de constantes (continuação)

Nome	Valor decimal	Valor Hexadecimal
MQCMD_INQUIRE_LOG	120	X'00000078'
MQCMD_INQUIRE_SECURITY	121	X'00000079'
MQCMD_INQUIRE_STG_CLASS	122	X'0000007A'
MQCMD_INQUIRE_SYSTEM	123	X'0000007B'
MQCMD_INQUIRE_THREAD	124	X'0000007C'
MQCMD_INQUIRE_TRACE	125	X'0000007D'
MQCMD_INQUIRE_USAGE	126	X'0000007E'
MQCMD_MOVE_Q	127	X'0000007F'
MQCMD_RECOVER_BSDFS	128	X'00000080'
MQCMD_RECOVER_CF_ESTRUTURA	129	X'00000081'
MQCMD_RESET_TPIPE	130	X'00000082'
MQCMD_RESOLVE_INDOUBT	131	X'00000083'
MQCMD_RESUME_Q_MGR	132	X'00000084'
MQCMD_REVERIFY_SECURITY	133	X'00000085'
MQCMD_SET_ARCHIVE	134	X'00000086'
MQCMD_SET_LOG	136	X'00000088'
MQCMD_SET_SYSTEM	137	X'00000089'
MQCMD_START_CMD_SERVER	138	X'0000008A'
MQCMD_START_Q_MGR	139	X'0000008B'
MQCMD_START_TRACE	140	X'0000008C'
MQCMD_STOP_CHANNEL_INIT	141	X'0000008D'
MQCMD_STOP_CHANNEL_LISTENER	142	X'0000008E'
MQCMD_STOP_CMD_SERVER	143	X'0000008F'
MQCMD_STOP_Q_MGR	144	X'00000090'
MQCMD_STOP_TRACE	145	X'00000091'
MQCMD_SUSPEND_Q_MGR	146	X'00000092'
MQCMD_INQUIRE_CF_STRUC_NAMES	147	X'00000093'
MQCMD_INQUIRE_STG_CLASS_NAMES	148	X'00000094'
MQCMD_CHANGE_SERVICE	149	X'00000095'
MQCMD_COPY_SERVICE	150	X'00000096'
MQCMD_CREATE_SERVICE	151	X'00000097'
MQCMD_DELETE_SERVICE	152	X'00000098'
MQCMD_INQUIRE_SERVICE	153	X'00000099'
MQCMD_INQUIRE_SERVICE_STATUS	154	X'0000009A'
MQCMD_START_SERVICE	155	X'0000009B'
MQCMD_STOP_SERVICE	156	X'0000009C'
MQCMD_DELETE_BUFFER_POOL	157	X'0000009D'
MQCMD_DELETE_PAGE_SET	158	X'0000009E'
MQCMD_CHANGE_BUFFER_POOL	159	X'0000009F'



Tabela 102. Valores de constantes (continuação)

Nome	Valor decimal	Valor Hexadecimal
MQCMD_CHANGE_PAGE_SET	160	X'000000A0'
MQCMD_INQUIRE_Q_MGR_STATUS	161	X'000000A1'
MQCMD_CREATE_LOG	162	X'000000A2'
MQCMD_STATISTICS_MQI	164	X'000000A4'
MQCMD_STATISTICS_Q	165	X'000000A5'
MQCMD_STATISTICS_CHANNEL	166	X'000000A6'
MQCMD_ACCOUNTING_MQI	167	X'000000A7'
MQCMD_ACCOUNTING_Q	168	X'000000A8'
MQCMD_INQUIRE_AUTH_SERVICE	169	X'000000A9'
MQCMD_CHANGE_TOPIC	170	X'000000AA'
MQCMD_COPY_TOPIC	171	X'000000AB'
MQCMD_CREATE_TOPIC	172	X'000000AC'
MQCMD_DELETE_TOPIC	173	X'000000AD'
MQCMD_INQUIRE_TOPIC	174	X'000000AE'
MQCMD_INQUIRE_TOPIC_NAMES	175	X'000000AF'
MQCMD_INQUIRE_SUBSCRIPTION	176	X'000000B0'
MQCMD_CREATE_SUBSCRIPTION	177	X'000000B1'
MQCMD_CHANGE_SUBSCRIPTION	178	X'000000B2'
MQCMD_DELETE_SUBSCRIPTION	179	X'000000B3'
MQCMD_COPY_SUBSCRIPTION	181	X'000000B5'
MQCMD_INQUIRE_SUB_STATUS	182	X'000000B6'
MQCMD_INQUIRE_TOPIC_STATUS	183	X'000000B7'
MQCMD_CLEAR_TOPIC_STRING	184	X'000000B8'
MQCMD_INQUIRE_PUBSUB_STATUS	185	X'000000B9'
MQCMD_PURGE_CHANNEL	195	X'000000C3'

### MQCMDI\_\* (Comando formata Valores de Informações de Comando)

Tabela 103. Valores de constantes

Nome	Valor decimal	Valor Hexadecimal
MQCMDI_CMDScope_ACCEPTED	1	X'00000001'
MQCMDI_CMDScope_GENERATED	2	X'00000002'
MQCMDI_CMDScope_COMPLETED	3	X'00000003'
MQCMDI_QSG_DISP_COMPLETED	4	X'00000004'
MQCMDI_COMMAND_ACCEPTED	5	X'00000005'
MQCMDI_CLUSTER_REQUEST_QUEUED	6	X'00000006'
MQCMDI_CHANNEL_INIT_STARTED	7	X'00000007'
MQCMDI_RECOVER_STARTED	11	X'0000000B'
MQCMDI_BACKUP_STARTED	12	X'0000000C'
MQCMDI_RECOVER_COMPLETED	13	X'0000000D'

<i>Tabela 103. Valores de constantes (continuação)</i>		
<b>Nome</b>	<b>Valor decimal</b>	<b>Valor Hexadecimal</b>
MQCMDI_SEC_TIMER_ZERO	14	X'0000000E'
MQCMDI_REFRESH_CONFIGURATION	16	X'00000010'
MQCMDI_SEC_SIGNOFF_ERROR	17	X'00000011'
MQCMDI_IMS_BRIDGE_SUSPENDED	18	X'00000012'
MQCMDI_DB2_SUSPENDED	19	X'00000013'
MQCMDI_DB2_OBSOLETE_MSGS	20	X'00000014'
MQCMDI_SEC_UPPERCASE	21	X'00000015'
MQCMDI_SEC_MIXEDCASE	22	X'00000016'

### **MQCMDL\_\* (Níveis de Comando)**

<i>Tabela 104. Nomes e valores de constantes</i>	
<b>Nome</b>	<b>Value</b>
MQCMDL_LEVEL_800	800
MQCMDL_LEVEL_801	801
MQCMDL_LEVEL_802	802
MQCMDL_LEVEL_900	900
MQCMDL_LEVEL_901	901
MQCMDL_LEVEL_902	902
MQCMDL_LEVEL_903	903
MQCMDL_LEVEL_904	904
MQCMDL_LEVEL_905	905
MQCMDL_LEVEL_910	910
MQCMDL_LEVEL_912	912
MQCMDL_LEVEL_913	913
MQCMDL_LEVEL_914	914
MQCMDL_LEVEL_915	915
MQCMDL_LEVEL_920	920
MQCMDL_LEVEL_921	921
MQCMDL_LEVEL_922	922
MQCMDL_LEVEL_923	923
MQCMDL_LEVEL_924	924
MQCMDL_LEVEL_925	925
MQCMDL_LEVEL_930	930
MQCMDL_LEVEL_931	931
MQCMDL_LEVEL_932	932

## MQCMHO\_\* (Criar opções e estrutura de manipulação de mensagem)..

### Criar estrutura de opções de manipulação de mensagens

Nome	Estrutura
MQCMHO_STRUC_ID	"CMHO"
MQCMHO_STRUC_ID_ARRAY	'C','M','H','O'

**Nota:** O símbolo ~ representa um único caractere em branco.

Nome	Valor decimal	Valor Hexadecimal
MQCMHO_VERSION_1	1	X'00000001'
MQCMHO_CURRENT_VERSION	1	X'00000001'

### Criar opções de manipulação de mensagem

Nome	Valor decimal	Valor Hexadecimal
MQCMHO_DEFAULT_VALIDATION	0	X'00000000'
MQCMHO_NO_VALIDATION	1	X'00000001'
MQCMHO_VALIDATE	2	X'00000002'
MQCMHO_NONE	0	X'00000000'

## MQCNO\_\* (Opções e estrutura de conexão).

### Conectar estrutura de opções

Nome	Estrutura
MQCNO_STRUC_ID	"CNO~"
MQCNO_STRUC_ID_ARRAY	'C','N','O','~'

**Nota:** O símbolo ~ representa um único caractere em branco.

Nome	Valor decimal	Valor Hexadecimal
MQCNO_VERSION_1	1	X'00000001'
MQCNO_VERSION_2	2	X'00000002'
MQCNO_VERSION_3	3	X'00000003'
MQCNO_VERSION_4	4	X'00000004'
MQCNO_VERSION_5	5	X'00000005'
MQCNO_CURRENT_VERSION	5	X'00000005'

## Opções de Conexão

*Tabela 110. Valores de constantes*

Nome	Valor decimal	Valor Hexadecimal
MQCNO_STANDARD_BINDING	0	X'00000000'
MQCNO_FASTPATH_BINDING	1	X'00000001'
MQCNO_SERIALIZE_CONN_TAG_Q_MGR	2	X'00000002'
MQCNO_SERIALIZE_CONN_TAG_QSG	4	X'00000004'
MQCNO_RESTRICT_CONN_TAG_Q_MGR	8	X'00000008'
MQCNO_RESTRICT_CONN_TAG_QSG	16	X'00000010'
MQCNO_HANDLE_SHARE_NONE	32	X'00000020'
MQCNO_HANDLE_SHARE_BLOCK	64	X'00000040'
MQCNO_HANDLE_SHARE_NO_BLOCK	128	X'00000080'
MQCNO_SHARED_BINDING	256	X'00000100'
MQCNO_ISOLATED_BINDING	512	X'00000200'
MQCNO_LOCAL_BINDING	1024	X'00000400'
MQCNO_CLIENT_BINDING	2048	X'00000800'
MQCNO_ACCOUNTING_MQI_ENABLED	4096	X'00001000'
MQCNO_ACCOUNTING_MQI_DISABLED	8192	X'00002000'
MQCNO_ACCOUNTING_Q_ENABLED	16384	X'00004000'
MQCNO_ACCOUNTING_Q_DISABLED	32768	X'00008000'
MQCNO_NO_CONV_SHARING	65536	X'00010000'
MQCNO_ALL_CONVS_SHARE	262144	X'00040000'
MQCNO_CD_FOR_OUTPUT_ONLY	524288	X'00080000'
MQCNO_USE_CD_SELECTION	1048576	X'00100000'
MQCNO_RECONNECT	16777216	X'01000000'
MQCNO_RECONNECT_AS_DEF	0	X'00000000'
MQCNO_RECONNECT_DISABLED	33554432	X'02000000'
MQCNO_RECONNECT_Q_MGR	67108864	X'04000000'
MQCNO_ACTIVITY_TRACE_ENABLED	134217728	X'08000000'
MQCNO_ACTIVITY_TRACE_DISABLED	268435456	X'10000000'
MQCNO_NONE	0	X'00000000'

## MQCO\_\* (Opções de fechamento)

*Tabela 111. Valores de constantes*

Nome	Valor decimal	Valor Hexadecimal
MQCO_IMMEDIATE	0	X'00000000'
MQCO_NONE	0	X'00000000'
MQCO_DELETE	1	X'00000001'
MQCO_DELETE_PURGE	2	X'00000002'
MQCO_KEEP_SUB	4	X'00000004'
MQCO_REMOVE_SUB	8	X'00000008'

Tabela 111. Valores de constantes (continuação)		
Nome	Valor decimal	Valor Hexadecimal
MQCO_QUIESCE	32	X'00000020'

### **MQCODL\_\* ( CICS comprimento de dados de saída do cabeçalho de informações)**

Tabela 112. Valores de constantes		
Nome	Valor decimal	Valor Hexadecimal
MQCODL_AS_INPUT	-1	X'FFFFFFFF'

### **MQCOMPRESS\_\* (Compactação de Canal)**

Tabela 113. Valores de constantes		
Nome	Valor decimal	Valor Hexadecimal
MQCOMPRESS_NOT_AVAILABLE	-1	X'FFFFFFFF'
MQCOMPRESS_NONE	0	X'00000000'
MQCOMPRESS_RLE	1	X'00000001'
MQCOMPRESS_ZLIBFAST	2	X'00000002'
MQCOMPRESS_ZLIBHIGH	4	X'00000004'
MQCOMPRESS_SYSTEM	8	X'00000008'
MQCOMPRESS_ANY	268435455	X'0FFFFFFFF'

### **MQCONNID\_\* (Identificador de Conexão).**

Tabela 114. Nomes e valores de constantes	
Nome	Value
MQCONNID_NONE	X'00...00' (24 nulos)
MQCONNID_NONE_ARRAY	'\0', '\0', ... (24 nulos)

### **MQCOPY\_\* (Opções de cópia de propriedade)**

Tabela 115. Valores de constantes		
Nome	Valor decimal	Valor Hexadecimal
MQCOPY_NONE	0	X'00000000'
MQCOPY_ALL	1	X'00000001'
MQCOPY_FORWARD	2	X'00000002'
MQCOPY_PUBLISH	4	X'00000004'
MQCOPY_REPLY	8	X'00000008'
MQCOPY_REPORT	16	X'00000010'
MQCOPY_DEFAULT	22	X'00000016'

### **MQCQT\_\* (Tipos de Fila de Cluster)**

Tabela 116. Valores de constantes		
Nome	Valor decimal	Valor Hexadecimal
MQCQT_LOCAL_Q	1	X'00000001'

Tabela 116. Valores de constantes (continuação)		
Nome	Valor decimal	Valor Hexadecimal
MQCQT_ALIAS_Q	2	X'00000002'
MQCQT_REMOTE_Q	3	X'00000003'
MQCQT_Q_MGR_ALIAS	4	X'00000004'

### **MQCRC\_\* (Códigos de Retorno do cabeçalho de informações do CICS)**

Tabela 117. Valores de constantes		
Nome	Valor decimal	Valor Hexadecimal
MQCRC_OK	0	X'00000000'
MQCRC_CICS_EXEC_ERROR	1	X'00000001'
MQCRC_MQ_API_ERROR	2	X'00000002'
MQCRC_BRIDGE_ERROR	3	X'00000003'
MQCRC_BRIDGE_ABEND	4	X'00000004'
MQCRC_APPLICATION_ABEND	5	X'00000005'
MQCRC_SECURITY_ERROR	6	X'00000006'
MQCRC_PROGRAM_NOT_AVAILABLE	7	X'00000007'
MQCRC_BRIDGE_TIMEOUT	8	X'00000008'
MQCRC_TRANSID_NOT_AVAILABLE	9	X'00000009'

### **MQCS\_\* (estado do consumidor de constantes MQCBC);**

Tabela 118. Valores de constantes		
Nome	Valor decimal	Valor Hexadecimal
MQCS_NONE	0	X'00000000'
MQCS_SUSPENDED_TEMPORARY	1	X'00000001'
MQCS_SUSPENDED_USER_ACTION	2	X'00000002'
MQCS_SUSPENDED	3	X'00000003'
MQCS_STOPPED	4	X'00000004'

### **MQCSC\_\* (Códigos de Início do cabeçalho de informação CICS)**

Tabela 119. Estruturas de constantes	
Nome	Estrutura
MQCSC_START	"S¬¬"
MQCSC_STARTDATA	"SD¬¬"
MQCSC_TERMINPUT	"TD¬¬"
MQCSC_NONE	"¬¬¬"
MQCSC_START_ARRAY	'S','¬','¬','¬'
MQCSC_STARTDATA_ARRAY	'S','D','¬','¬'
MQCSC_TERMINPUT_ARRAY	'T','D','¬','¬'
MQCSC_NONE_ARRAY	'¬','¬','¬','¬'

**Nota:** O símbolo ¬ representa um único caractere em branco.

## MQCSP\_\* (Estrutura de parâmetros de segurança de conexão e Tipos de autenticação).

### Estrutura dos parâmetros de segurança da conexão

Nome	Estrutura
MQCSP_STRUC_ID	"CSP↵"
MQCSP_STRUC_ID_ARRAY	'C', 'S', 'P', '↵'

**Nota:** O símbolo ↵ representa um único caractere em branco.

Nome	Valor decimal	Valor Hexadecimal
MQCSP_VERSION_1	1	X'00000001'
MQCSP_VERSION_2	2	X'00000002'
MQCSP_VERSION_3	3	X'00000003'
MQCSP_CURRENT_VERSION	3	X'00000003'

### Parâmetros de segurança de conexão

Nome	Valor decimal	Valor Hexadecimal
MQCSP_AUTH_NONE	0	X'00000000'
MQCSP_AUTH_USER_ID_AND_PWD	1	X'00000001'
MQCSP_AUTH_ID_TOKEN	2	X'00000002'

### MQCSRV\_\* (Opções do Servidor de Comandos)..

Nome	Valor decimal	Valor Hexadecimal
MQCSRV_CONVERT_NO	0	X'00000000'
MQCSRV_CONVERT_YES	1	X'00000001'
MQCSRV_DLQ_NO	0	X'00000000'
MQCSRV_DLQ_YES	1	X'00000001'

### MQCT\_\* (Tag de conexão do gerenciador de filas)

Nome	Value
MQCT_NONE	X'00...00' (128 nulos)
MQCT_NONE_ARRAY	'\0', '\0', ... (128 nulos)

## MQCTES\_\* ( CICS Status de Encerramento da Tarefa do cabeçalho de informações)

*Tabela 125. Valores de constantes*

Nome	Valor decimal	Valor Hexadecimal
MQCTES_NOSYNC	0	X'00000000'
MQCTES_COMMIT	256	X'00000100'
MQCTES_BACKOUT	4352	X'00001100'
MQCTES_ENDTASK	65536	X'00010000'

## MQCTLO\_\* (estrutura de opções MQCTL e Opções de Controle do Consumidor)

### Estrutura de opções MQCTL

*Tabela 126. Estruturas de constantes*

Nome	Estrutura
MQCTLO_STRUC_ID	"CTLO"
MQCTLO_STRUC_ID_ARRAY	'C','T','L','O'

**Nota:** O símbolo ~ representa um único caractere em branco.

*Tabela 127. Valores de constantes*

Nome	Valor decimal	Valor Hexadecimal
MQCTLO_VERSION_1	1	X'00000001'
MQCTLO_CURRENT_VERSION	1	X'00000001'

### Opções MQCTL Opções de Controle do Consumidor

*Tabela 128. Valores de constantes*

Nome	Valor decimal	Valor Hexadecimal
MQCTLO_NONE	0	X'00000000'
MQCTLO_THREAD_AFFINITY	1	X'00000001'
MQCTLO_FAIL_IF QUIESCING	8192	X'00002000'

## MQCUOWC\_\* (Controles de Unidade de Trabalho do cabeçalho de informações CICS)

*Tabela 129. Valores de constantes*

Nome	Valor decimal	Valor Hexadecimal
MQCUOWC_ONLY	273	X'00000111'
MQCUOWC_CONTINUE	65536	X'00010000'
MQCUOWC_FIRST	17	X'00000011'
MQCUOWC_MIDDLE	16	X'00000010'
MQCUOWC_LAST	272	X'00000110'
MQCUOWC_COMMIT	256	X'00000100'
MQCUOWC_BACKOUT	4352	X'00001100'



## MQCXP\_\* (Estrutura do parâmetro de saída do canal)

Tabela 130. Estruturas de constantes	
Nome	Estrutura
MQCXP_STRUC_ID	"CXP¬"
MQCXP_STRUC_ID_ARRAY	'C', 'X', 'P', '¬'

**Nota:** O símbolo ¬ representa um único caractere em branco.

Tabela 131. Valores de constantes		
Nome	Valor decimal	Valor Hexadecimal
MQCXP_VERSION_1	1	X'00000001'
MQCXP_VERSION_2	2	X'00000002'
MQCXP_VERSION_3	3	X'00000003'
MQCXP_VERSION_4	4	X'00000004'
MQCXP_VERSION_5	5	X'00000005'
MQCXP_VERSION_6	6	X'00000006'
MQCXP_VERSION_7	7	X'00000007'
MQCXP_VERSION_8	8	X'00000008'
MQCXP_VERSION_9	9	X'00000009'
MQCXP_CURRENT_VERSION	9	X'00000009'

## MQDC\_\* (Classe de Destino).

Tabela 132. Valores de constantes		
Nome	Valor decimal	Valor Hexadecimal
MQDC_MANAGED	1	X'00000001'
MQDC_PROVIDED	2	X'00000002'

## MQDCC\_\* (Opções de Conversão e Máscaras e Fatores)

### Opções de conversão

Tabela 133. Valores de constantes		
Nome	Valor decimal	Valor Hexadecimal
MQDCC_DEFAULT_CONVERSION	1	X'00000001'
MQDCC_FILL_TARGET_BUFFER	2	X'00000002'
MQDCC_INT_DEFAULT_CONVERSION	4	X'00000004'
MQDCC_SOURCE_ENC_NATIVE	(value differs by platform or version)	(value differs by platform or version)
MQDCC_SOURCE_ENC_NORMAL	16	X'00000010'
MQDCC_SOURCE_ENC_REVERTIDO	32	X'00000020'
MQDCC_SOURCE_ENC_UNDEFINED	0	X'00000000'
MQDCC_TARGET_ENC_NATIVE	(value differs by platform or version)	(value differs by platform or version)
MQDCC_TARGET_ENC_NORMAL	256	X'00000100'
MQDCC_TARGET_ENC_REVERTIDO	512	X'00000200'

<i>Tabela 133. Valores de constantes (continuação)</i>		
Nome	Valor decimal	Valor Hexadecimal
MQDCC_TARGET_ENC_UNDEFINED	0	X'00000000'
MQDCC_NONE	0	X'00000000'

### Máscaras e Fatores de Opções de Conversão

<i>Tabela 134. Valores de constantes</i>		
Nome	Valor decimal	Valor Hexadecimal
MQDCC_SOURCE_ENC_MASK	240	X'000000F0'
MQDCC_TARGET_ENC_MASK	3840	X'00000F00'
MQDCC_SOURCE_ENC_FACTOR	16	X'00000010'
MQDCC_TARGET_ENC_FACTOR	256	X'00000100'

### MQDELO\_\* (Opções de Exclusão de Publicação / Assinatura)

<i>Tabela 135. Valores de constantes</i>		
Nome	Valor decimal	Valor Hexadecimal
MQDELO_NONE	0	X'00000000'
MQDELO_LOCAL	4	X'00000004'

### MQDH\_\* (Estrutura do cabeçalho de distribuição)

<i>Tabela 136. Estruturas de constantes</i>	
Nome	Estrutura
MQDH_STRUC_ID	"DH--"
MQDH_STRUC_ID_ARRAY	'D','H',' ',' '

**Nota:** O símbolo - representa um único caractere em branco.

<i>Tabela 137. Valores de constantes</i>		
Nome	Valor decimal	Valor Hexadecimal
MQDH_VERSION_1	1	X'00000001'
MQDH_CURRENT_VERSION	1	X'00000001'

### MQDHF\_\* (Sinalizadores do cabeçalho de distribuição)

<i>Tabela 138. Valores de constantes</i>		
Nome	Valor decimal	Valor Hexadecimal
MQDHF_NEW_MSG_IDS	1	X'00000001'
MQDHF_NONE	0	X'00000000'

### MQDISCONNECT\_\* (Tipos de Desconexão do Formato de Comandos)

<i>Tabela 139. Valores de constantes</i>		
Nome	Valor decimal	Valor Hexadecimal
MQDISCONNECT_NORMAL	0	X'00000000'
MQDISCONNECT_IMPLICIT	1	X'00000001'

Tabela 139. Valores de constantes (continuação)		
Nome	Valor decimal	Valor Hexadecimal
MQDISCONNECT_Q_MGR	2	X'00000002'

## MQDL\_\* (Listas de Distribuição)

Tabela 140. Valores de constantes		
Nome	Valor decimal	Valor Hexadecimal
MQDL_SUPPORTED	1	X'00000001'
MQDL_NOT_SUPPORTED	0	X'00000000'

## MQDLH\_\* (estrutura do cabeçalho de Devoluções)

Tabela 141. Estruturas de constantes	
Nome	Estrutura
MQDLH_STRUC_ID	"DLH↵"
MQDLH_STRUC_ID_ARRAY	'D', 'L', 'H', '↵'

**Nota:** O símbolo ↵ representa um único caractere em branco.

MQDLH_VERSION_1	1	X'00000001'
MQDLH_CURRENT_VERSION	1	X'00000001'

## MQDLV\_\* (Entrega de Mensagem Persistente / Não persistente)

Tabela 142. Valores de constantes		
Nome	Valor decimal	Valor Hexadecimal
MQDLV_AS_PARENT	0	X'00000000'
MQDLV_ALL	1	X'00000001'
MQDLV_ALL_DUR	2	X'00000002'
MQDLV_ALL_AVAIL	3	X'00000003'

## MQDMHO\_\* (Excluir opções de manipulação de mensagem e estrutura).

### Excluir estrutura de opções de manipulação de mensagem

Tabela 143. Estruturas de constantes	
Nome	Estrutura
MQDMHO_STRUC_ID	"DMHO"
MQDMHO_STRUC_ID_ARRAY	'D', 'M', 'H', 'O'

**Nota:** O símbolo ↵ representa um único caractere em branco.

Tabela 144. Valores de constantes		
Nome	Valor decimal	Valor Hexadecimal
MQDMHO_VERSION_1	1	X'00000001'
MQDMHO_CURRENT_VERSION	1	X'00000001'

## Excluir opções de manipulação de mensagens

Tabela 145. Valores de constantes		
Nome	Valor decimal	Valor Hexadecimal
MQDMHO_NONE	0	X'00000000'

## MQDMPO\_\* (Excluir opções e estrutura da propriedade de mensagem)

### Excluir estrutura de opções de propriedade de mensagens

Tabela 146. Estruturas de constantes	
Nome	Estrutura
MQDMPO_STRUC_ID	"DMPO"
MQDMPO_STRUC_ID_ARRAY	'D', 'M', 'P', 'O'

**Nota:** O símbolo ~ representa um único caractere em branco.

Tabela 147. Valores de constantes		
Nome	Valor decimal	Valor Hexadecimal
MQDMPO_VERSION_1	1	X'00000001'
MQDMPO_CURRENT_VERSION	1	X'00000001'

## Excluir Opções de Propriedade de Mensagem

Tabela 148. Valores de constantes		
Nome	Valor decimal	Valor Hexadecimal
MQDMPO_DEL_FIRST	0	X'00000000'
MQDMPO_DEL_PROP_UNDER_CURSOR	1	X'00000001'
MQDMPO_NONE	0	X'00000000'

## MQDNSWLM\_\* (DNS WLM)

Tabela 149. Valores de constantes		
Nome	Valor decimal	Valor Hexadecimal
MQDNSWLM_NO	0	X'00000000'
MQDNSWLM_YES	1	X'00000001'

## MQDT\_\* (Tipos de Destino).

Tabela 150. Valores de constantes		
Nome	Valor decimal	Valor Hexadecimal
MQDT_APPL	1	X'00000001'
MQDT_BROKER	2	X'00000002'

## MQDXP\_\* (Estrutura do parâmetro de saída de conversão)

Tabela 151. Estruturas de constantes	
Nome	Estrutura
MQDXP_STRUC_ID	"DXP~"

Tabela 151. Estruturas de constantes (continuação)	
Nome	Estrutura
MQDXP_STRUC_ID_ARRAY	'D', 'X', 'P', ' '

**Nota:** O símbolo – representa um único caractere em branco.

Tabela 152. Valores de constantes		
Nome	Valor decimal	Valor Hexadecimal
MQDXP_VERSION_1	1	X'00000001'
MQDXP_VERSION_2	2	X'00000002'
MQDXP_CURRENT_VERSION	2	X'00000002'

### MQEC\_\* (Valores de sinal)

Tabela 153. Valores de constantes		
Nome	Valor decimal	Valor Hexadecimal
MQEC_MSG_CHEGOU	2	X'00000002'
MQEC_WAIT_INTERVAL_EXPIRED	3	X'00000003'
MQEC_WAIT_CANCELADO	4	X'00000004'
MQEC_Q_MGR QUIESCING	5	X'00000005'
MQEC_CONNECTION QUIESCING	6	X'00000006'

### MQEI\_\* (expiração)

Tabela 154. Valores de constantes		
Nome	Valor decimal	Valor Hexadecimal
MQEI_UNLIMITED	-1	X'FFFFFFFF'

### MQENC\_\* (Codificação)

### MQENC\_\* (Codificação)

Tabela 155. Valores de constantes por plataforma			
Nome	Plataforma	Valor decimal	Valor Hexadecimal
MQENC_NATIVE	IBM i	273	X'00000111'
	Linux	546	X'00000222'
	Linux no SPARC	273	X'00000111'
	Linux em x86	546	X'00000222'
	AIX and Linux	273	X'00000111'
	Windows	546	X'00000222'
	Micro Focus COBOL no Windows	17	X'00000011'
	z/OS	785	X'00000311'

## MQENC\_\* (Máscaras de Codificação)

Tabela 156. Valores de constantes		
Nome	Valor decimal	Valor Hexadecimal
MQENC_INTEGER_MASK	15	X'0000000F'
MQENC_DECIMAL_MASK	240	X'000000F0'
MQENC_FLOAT_MASK	3840	X'00000F00'
MQENC_RESERVED_MASK	-4096	X'FFFFFF000'

## MQENC\_\* (codificações para números inteiros binários)

Tabela 157. Valores de constantes		
Nome	Valor decimal	Valor Hexadecimal
MQENC_INTEGER_UNDEFINED	0	X'00000000'
MQENC_INTEGER_NORMAL	1	X'00000001'
MQENC_INTEGER_REVERSED	2	X'00000002'

## MQENC\_\* (Codificações para Números Inteiros Decimais Compactados)

Tabela 158. Valores de constantes		
Nome	Valor decimal	Valor Hexadecimal
MQENC_DECIMAL_UNDEFINED	0	X'00000000'
MQENC_DECIMAL_NORMAL	16	X'00000010'
MQENC_DECIMAL_REVERSED	32	X'00000020'

## MQENC\_\* (Codificações para Números de Ponto Flutuante)

Tabela 159. Valores de constantes		
Nome	Valor decimal	Valor Hexadecimal
MQENC_FLOAT_UNDEFINED	0	X'00000000'
MQENC_FLOAT_IEEE_NORMAL	256	X'00000100'
MQENC_FLOAT_IEEE_REVERSED	512	X'00000200'
MQENC_FLOAT_S390	768	X'00000300'
MQENC_FLOAT_TNS	1024	X'00000400'

## MQEPH\_\* (Estrutura do cabeçalho do formato de comando integrado e Sinalizações)

### Estrutura do cabeçalho do formato de comando integrado

Tabela 160. Estruturas de constantes	
Nome	Estrutura
MQEPH_STRUC_ID	"EPH~"
MQEPH_STRUC_ID_ARRAY	'E', 'P', 'H', '~'

**Nota:** O símbolo ~ representa um único caractere em branco.

<i>Tabela 161. Valores de constantes</i>		
Nome	Valor decimal	Valor Hexadecimal
MQEPH_STRUC_LENGTH_FIXED	68	X'00000044'
MQEPH_VERSION_1	1	X'00000001'
MQEPH_CURRENT_VERSION	1	X'00000001'

### Sinalizadores do cabeçalho do formato de comando integrado

<i>Tabela 162. Valores de constantes</i>		
Nome	Valor decimal	Valor Hexadecimal
MQEPH_NONE	0	X'00000000'
MQEPH_CCSID_EMBEDDED	1	X'00000001'

### MQET\_\* (Tipos de Escape do Formato de Comando)

<i>Tabela 163. Valores de constantes</i>		
Nome	Valor decimal	Valor Hexadecimal
MQET_MQSC	1	X'00000001'

### MQEVO\_\* (Origens de Eventos do Formato de Comando)..

<i>Tabela 164. Valores de constantes</i>		
Nome	Valor decimal	Valor Hexadecimal
MQEVO_OTHER	0	X'00000000'
MQEVO_CONSOLE	1	X'00000001'
MQEVO_INIT	2	X'00000002'
MQEVO_MSG	3	X'00000003'
MQEVO_MQSET	4	X'00000004'
MQEVO_INTERNAL	5	X'00000005'
MQEVO_MQSUB	6	X'00000006'
MQEVO_CTLMSG	7	X'00000007'
MQEVO_REST	8	X'00000008'

### MQEVR\_\* (Gravação de Eventos no formato de comando)..

<i>Tabela 165. Valores de constantes</i>		
Nome	Valor decimal	Valor Hexadecimal
MQEVR_DISABLED	0	X'00000000'
MQEVR_ENABLED	1	X'00000001'
MQEVR_EXCEPTION	2	X'00000002'
MQEVR_NO_DISPLAY	3	X'00000003'

### MQEXPI\_\* (intervalo de varredura de expiração)

<i>Tabela 166. Valores de constantes</i>		
Nome	Valor decimal	Valor Hexadecimal
MQEXPI_OFF	0	X'00000000'

## MQFB\_ \* (Valores de feedback)

*Tabela 167. Valores de constantes*

Nome	Valor decimal	Valor Hexadecimal
MQFB_NONE	0	X'00000000'
MQFB_SYSTEM_FIRST	1	X'00000001'
MQFB_QUIT	256	X'00000100'
MQFB_EXPIRATION	258	X'00000102'
MQFB_COA	259	X'00000103'
MQFB_COD	260	X'00000104'
MQFB_CHANNEL_COMPLETED	262	X'00000106'
MQFB_CHANNEL_FAIL_RETRY	263	X'00000107'
MQFB_CHANNEL_FAIL	264	X'00000108'
MQFB_APPL_CANNOT_BE_STARTED	265	X'00000109'
MQFB_TM_ERROR	266	X'0000010A'
MQFB_APPL_TYPE_ERROR	267	X'0000010B'
MQFB_STOPPED_BY_MSG_EXIT	268	X'0000010C'
MQFB_ACTIVITY	269	X'0000010D'
MQFB_XMIT_Q_MSG_ERROR	271	X'0000010F'
MQFB_PAN	275	X'00000113'
MQFB_NAN	276	X'00000114'
MQFB_STOPPED_BY_CHAD_EXIT	277	X'00000115'
MQFB_STOPPED_BY_PUBSUB_EXIT	279	X'00000117'
MQFB_NOT_A_REPOSITORY_MSG	280	X'00000118'
MQFB_BIND_OPEN_CLUSRCVR_DEL	281	X'00000119'
MQFB_MAX_ACTIVIDADES	282	X'0000011A'
MQFB_NOT_FORWARDED	283	X'0000011B'
MQFB_NOT_ENTREGUE	284	X'0000011C'
MQFB_UNSUPPORTED_FORWARDING	285	X'0000011D'
MQFB_UNSUPPORTED_DELIVERY	286	X'0000011E'
MQFB_DATA_LENGTH_ZERO	291	X'00000123'
MQFB_DATA_LENGTH_NEGATIVE	292	X'00000124'
MQFB_DATA_LENGTH_TOO_BIG	293	X'00000125'
MQFB_BUFFER_OVERFLOW	294	X'00000126'
MQFB_LENGTH_OFF_BY_ONE	295	X'00000127'
MQFB_IIH_ERROR	296	X'00000128'
MQFB_NOT_AUTHORIZED_FOR_IMS	298	X'0000012A'
MQFB_IMS_ERROR	300	X'0000012C'
MQFB_IMS_FIRST	301	X'0000012D'
MQFB_IMS_LAST	399	X'0000018F'
MQFB_CICS_INTERNAL_ERROR	401	X'00000191'
MQFB_CICS_NOT_AUTHORIZED	402	X'00000192'



Tabela 167. Valores de constantes (continuação)

Nome	Valor decimal	Valor Hexadecimal
MQFB_CICS_BRIDGE_FAILURE	403	X'00000193'
MQFB_CICS_CORREL_ID_ERROR	404	X'00000194'
MQFB_CICS_CCSDID_ERROR	405	X'00000195'
MQFB_CICS_ENCODING_ERROR	406	X'00000196'
MQFB_CICS_CIH_ERROR	407	X'00000197'
MQFB_CICS_UOW_ERROR	408	X'00000198'
MQFB_CICS_COMMAREA_ERROR	409	X'00000199'
MQFB_CICS_APPL_NOT_STARTED	410	X'0000019A'
MQFB_CICS_APPL_ABENDED	411	X'0000019B'
MQFB_CICS_DLQ_ERROR	412	X'0000019C'
MQFB_CICS_UOW_BACKED_OUT	413	X'0000019D'
MQFB_PUBLICATIONS_ON_REQUEST	501	X'000001F5'
MQFB_SUBSCRIBER_IS_PUBLISHER	502	X'000001F6'
MQFB_MSG_SCOPE_MISMATCH	503	X'000001F7'
MQFB_SELECTOR_MISMATCH	504	X'000001F8'
MQFB_IMS_NACK_1A_REASON_FIRST	600	X'00000258'
MQFB_IMS_NACK_1A_REASON_LAST	855	X'00000357'
MQFB_SYSTEM_LAST	65535	X'0000FFFF'
MQFB_APPL_FIRST	65536	X'00010000'
MQFB_APPL_LAST	999999999	X'3B9AC9FF'

### MQFC\_\* (Formato de Comando Forçar Opções)

Tabela 168. Valores de constantes

Nome	Valor decimal	Valor Hexadecimal
MQFC_YES	1	X'00000001'
MQFC_NO	0	X'00000000'

### MQFMT\_\* (formatos)

Tabela 169. Nomes e valores de constantes

Nome	Value
MQFMT_NONE	"_ _ _ _ _ _ _ _"
MQFMT_ADMIN	"MQADMIN_"
MQFMT_CHANNEL_COMPLETED	"MQCHCOM_"
MQFMT_CICS	"MQCICS_ _"
MQFMT_COMMAND_1	"MQCMD1_ _"
MQFMT_COMMAND_2	"MQCMD2_ _"
MQFMT_DEAD_LETTER_HEADER	"MQDEAD_ _"
MQFMT_DIST_HEADER	"MQHDIST_"
MQFMT_EMBEDDED_PCF	"MQHEPCF_"

Tabela 169. Nomes e valores de constantes (continuação)

Nome	Value
MQFMT_EVENT	"MQEVENT~"
MQFMT_IMS	"MQIMS~~~"
MQFMT_IMS_VAR_STRING	"MQIMSVS~"
MQFMT_MD_EXTENSION	"MQHMDE~~"
MQFMT_PCF	"MQPCF~~~"
MQFMT_REF_MSG_HEADER	"MQHREF~~"
MQFMT_RF_HEADER	"MQHRF~~~"
MQFMT_RF_HEADER_1	"MQHRF~~~"
MQFMT_RF_HEADER_2	"MQHRF2~~"
MQFMT_STRING	"MQSTR~~~"
MQFMT_TRIGGER	"MQTRIG~~"
MQFMT_WORK_INFO_HEADER	"MQHWIH~~"
MQFMT_XMIT_Q_HEADER	"MQXMIT~~"
MQFMT_NONE_ARRAY	'~','~','~','~','~','~','~','~','~'
MQFMT_ADMIN_ARRAY	'M','Q','A','D','M','I','N','~'
MQFMT_CHANNEL_COMPLETED_ARRAY	'M','Q','C','H','C','O','M','~'
MQFMT_CICS_ARRAY	'M','Q','C','I','C','S','~','~'
MQFMT_COMMAND_1_ARRAY	'M','Q','C','M','D','1','~','~'
MQFMT_COMMAND_2_ARRAY	'M','Q','C','M','D','2','~','~'
MQFMT_DEAD_LETTER_HEADER_ARRAY	'M','Q','D','E','A','D','~','~'
MQFMT_DIST_HEADER_ARRAY	'M','Q','H','D','I','S','T','~'
MQFMT_EMBEDDED_PCF_ARRAY	'M','Q','H','E','P','C','F','~'
MQFMT_EVENT_ARRAY	'M','Q','E','V','E','N','T','~'
MQFMT_IMS_ARRAY	'M','Q','I','M','S','~','~','~'
MQFMT_IMS_VAR_STRING_ARRAY	'M','Q','I','M','S','V','S','~'
MQFMT_MD_EXTENSION_ARRAY	'M','Q','H','M','D','E','~','~'
MQFMT_PCF_ARRAY	'M','Q','P','C','F','~','~','~'
MQFMT_REF_MSG_HEADER_ARRAY	'M','Q','H','R','E','F','~','~'
MQFMT_RF_HEADER_ARRAY	'M','Q','H','R','F','~','~','~'
MQFMT_RF_HEADER_1_ARRAY	'M','Q','H','R','F','~','~','~'
MQFMT_RF_HEADER_2_ARRAY	'M','Q','H','R','F','2','~','~'
MQFMT_STRING_ARRAY	'M','Q','S','T','R','~','~','~'
MQFMT_TRIGGER_ARRAY	'M','Q','T','R','I','G','~','~'
MQFMT_WORK_INFO_HEADER_ARRAY	'M','Q','H','W','I','H','~','~'
MQFMT_XMIT_Q_HEADER_ARRAY	'M','Q','X','M','I','T','~','~'

**Nota:** O símbolo ~ representa um único caractere em branco.

## MQFUN\_\* (Tipos de função do aplicativo).

Tabela 170. Valores de constantes		
Nome	Valor decimal	Valor Hexadecimal
MQFUN_TYPE_UNKNOWN	0	X'00000000'
MQFUN_TYPE_JVM	1	X'00000001'
MQFUN_TYPE_PROGRAM	2	X'00000002'
MQFUN_TYPE_PROCEDURE	3	X'00000003'
MQFUN_TYPE_USERDEF	4	X'00000004'
MQFUN_TYPE_COMMAND	5	X'00000005'

## MQGA\_\* (Seletores de Atributo de Grupo).

Tabela 171. Valores de constantes		
Nome	Valor decimal	Valor Hexadecimal
MQGA_FIRST	8001	X'00001F41'
MQGA_LAST	9000	X'00002328'

## MQGACF\_\* (Tipos de Parâmetro do Grupo de Formato de Comandos)..

Tabela 172. Valores de constantes		
Nome	Valor decimal	Valor Hexadecimal
MQGACF_FIRST	8001	X'00001F41'
MQGACF_COMMAND_CONTEXT	8001	X'00001F41'
MQGACF_COMMAND_DATA	8002	X'00001F42'
MQGACF_TRACE_ROUTE	8003	X'00001F43'
MQGACF_OPERATION	8004	X'00001F44'
MQGACF_ACTIVITY	8005	X'00001F45'
MQGACF_EMBEDDED_MQMD	8006	X'00001F46'
MQGACF_MESSAGE	8007	X'00001F47'
MQGACF_MQMD	8008	X'00001F48'
MQGACF_VALUE_NAMING	8009	X'00001F49'
MQGACF_Q_ACCOUNTING_DATA	8010	X'00001F4A'
MQGACF_Q_STATISTICS_DATA	8011	X'00001F4B'
MQGACF_CHL_STATISTICS_DATA	8012	X'00001F4C'
MQGACF_LAST_USED	8012	X'00001F4C'

## MQGI\_\* (Identificador do grupo)

Tabela 173. Nomes e valores de constantes	
Nome	Value
MQGI_NONE	X'00...00' (24 nulos)
MQGI_NONE_ARRAY	'\0', '\0', ... (24 nulos)

## MQGMO\_\* (Obter opções de mensagem e estrutura).

### Obter estrutura de opções de mensagem

Nome	Estrutura
MQGMO_STRUC_ID	"GMO~"
MQGMO_STRUC_ID_ARRAY	'G','M','O','~'

**Nota:** O símbolo ~ representa um único caractere em branco.

Nome	Valor decimal	Valor Hexadecimal
MQGMO_VERSION_1	1	X'00000001'
MQGMO_VERSION_2	2	X'00000002'
MQGMO_VERSION_3	3	X'00000003'
MQGMO_VERSION_4	4	X'00000004'
MQGMO_CURRENT_VERSION	4	X'00000004'

### Obter opções de mensagem

Nome	Valor decimal	Valor Hexadecimal
MQGMO_WAIT	1	X'00000001'
MQGMO_NO_WAIT	0	X'00000000'
MQGMO_SET_SIGNAL	8	X'00000008'
MQGMO_FAIL_IF QUIESCING	8192	X'00002000'
MQGMO_SYNCPOINT	2	X'00000002'
MQGMO_SYNCPOINT_IF_PERSISTENT	4096	X'00001000'
MQGMO_NO_SYNCPOINT	4	X'00000004'
MQGMO_MARK_SKIP_BACKOUT	128	X'00000080'
MQGMO_BROWSE_FIRST	16	X'00000010'
MQGMO_BROWSE_NEXT	32	X'00000020'
MQGMO_BROWSE_MSG_UNDER_CURSOR	2048	X'00008000'
MQGMO_BROWSE_HANDLE	17825808	X'01100010'
MQGMO_BROWSE_CO_OP	18874384	X'01200010'
MQGMO_MSG_UNDER_CURSOR	256	X'00000100'
MQGMO_LOCK	512	X'00000200'
MQGMO_UNLOCK	1024	X'00000400'
MQGMO_ACCEPT_TRUNCATED_MSG	64	X'00000040'
MQGMO_CONVERT	16384	X'00004000'
MQGMO_LOGICAL_ORDER	32768	X'00008000'
MQGMO_COMPLETE_MSG	65536	X'00010000'
MQGMO_ALL_MSGS_AVAILABLE	131072	X'00020000'
MQGMO_ALL_SEGMENTS_AVAILABLE	262144	X'00040000'

<i>Tabela 176. Valores de constantes (continuação)</i>		
<b>Nome</b>	<b>Valor decimal</b>	<b>Valor Hexadecimal</b>
MQGMO_MARK_BROWSE_HANDLE	1048576	X'00100000'
MQGMO_MARK_BROWSE_CO_OP	2097152	X'00200000'
MQGMO_UNMARK_BROWSE_CO_OP	4194304	X'00400000'
MQGMO_UNMARK_BROWSE_HANDLE	8388608	X'00800000'
MQGMO_UNMARKED_BROWSE_MSG	16777216	X'01000000'
MQGMO_PROPERTIES_FORCE_MQRFH2	33554432	X'02000000'
MQGMO_NO_PROPERTIES	67108864	X'04000000'
MQGMO_PROPERTIES_IN_HANDLE	134217728	X'08000000'
MQGMO_PROPERTIES_COMPATIBILITY	268435456	X'10000000'
MQGMO_PROPERTIES_AS_Q_DEF	0	X'00000000'
MQGMO_NONE	0	X'00000000'

### **MQGS\_\* (Status do Grupo).**

<i>Tabela 177. Nomes e valores de constantes</i>	
<b>Nome</b>	<b>Value</b>
MQGS_NOT_IN_GROUP	'-'
MQGS_MSG_IN_GROUP	'G'
MQGS_LAST_MSG_IN_GROUP	'L'

**Nota:** O símbolo - representa um único caractere em branco.

### **MQHA\_\* (Manipular Seletores)**

<i>Tabela 178. Valores de constantes</i>		
<b>Nome</b>	<b>Valor decimal</b>	<b>Valor Hexadecimal</b>
MQHA_FIRST	4001	X'00000FA1'
MQHA_BAG_HANDLE	4001	X'00000FA1'
MQHA_LAST_USED	4001	X'00000FA1'
MQHA_LAST	6000	X'00001770'

### **MQHB\_\* (Alças de Saco)**

<i>Tabela 179. Valores de constantes</i>		
<b>Nome</b>	<b>Valor decimal</b>	<b>Valor Hexadecimal</b>
MQHB_UNUSABLE_HBAG	-1	X'FFFFFFFF'
MQHB_NONE	-2	X'FFFFFFFE'

### **MQHC\_\* (Identificadores de Conexão).**

<i>Tabela 180. Valores de constantes</i>		
<b>Nome</b>	<b>Valor decimal</b>	<b>Valor Hexadecimal</b>
MQHC_DEF_HCONN	0	X'00000000'
MQHC_UNUSABLE_HCONN	-1	X'FFFFFFFF'

Tabela 180. Valores de constantes (continuação)		
Nome	Valor decimal	Valor Hexadecimal
MQHC_UNASSOCIATED_HCONN	-3	X'FFFFFFFFD'

### MQHM\_\* (Identificador de mensagem).

Tabela 181. Valores de constantes		
Nome	Valor decimal	Valor Hexadecimal
MQHM_UNUSABLE_HMSG	-1	X'FFFFFFFF'
MQHM_NONE	0	X'00000000'

### MQHO\_\* (Identificador de Objeto).

Tabela 182. Valores de constantes		
Nome	Valor decimal	Valor Hexadecimal
MQHO_UNUSABLE_HOBJ	-1	X'FFFFFFFF'
MQHO_NONE	0	X'00000000'

### MQHSTATE\_\* (Estados do Manipulador do formato de comando)...

Tabela 183. Valores de constantes		
Nome	Valor decimal	Valor Hexadecimal
MQHSTATE_INACTIVE	0	X'00000000'
MQHSTATE_ACTIVE	1	X'00000001'

### MQIA\_\* (Seletores de atributo de número inteiro)

Tabela 184. Valores de constantes		
Nome	Valor decimal	Valor Hexadecimal
MQIA_ACCOUNTING_CONN_OVERRIDE	136	X'00000088'
MQIA_ACCOUNTING_INTERVAL	135	X'00000087'
MQIA_ACCOUNTING_MQI	133	X'00000085'
MQIA_ACCOUNTING_Q	134	X'00000086'
MQIA_ACTIVE_CHANNELS	100	X'00000064'
MQIA_ACTIVITY_CONN_OVERRIDE	239	X'000000EF'
MQIA_ACTIVITY_RECORDING	138	X'0000008A'
MQIA_ACTIVITY_TRACE	240	X'000000F0'
MQIA_ADOPTNEWMCA_CHECK	102	X'00000066'
MQIA_ADOPTNEWMCA_INTERVAL	104	X'00000068'
MQIA_ADOPTNEWMCA_TYPE	103	X'00000067'
MQIA_ADOPT_CONTEXT	260	X'00000104'
MQIA_ADVANCED_CAPABILITY	273	X'00000111'
MQIA_AMQP_CAPABILITY	265	X'00000109'
MQIA_APPL_TYPE	1	X'00000001'
MQIA_ARCHIVE	60	X'0000003C'

Tabela 184. Valores de constantes (continuação)

Nome	Valor decimal	Valor Hexadecimal
MQIA_AUTHENTICATION_FAIL_DELAY	259	X'00000103'
MQIA_AUTHENTICATION_METHOD	266	X'0000010A'
MQIA_AUTH_INFO_TYPE	66	X'00000042'
MQIA_AUTHORITY_EVENT	47	X'0000002F'
MQIA_AUTO_REORG_INTERVAL	174	X'000000AE'
MQIA_AUTO_REORGANIZATION	173	X'000000AD'
MQIA_BACKOUT_THRESHOLD	22	X'00000016'
MQIA_BASE_TYPE	193	X'000000C1'
MQIA_BATCH_INTERFACE_AUTO	86	X'00000056'
MQIA_BRIDGE_EVENT	74	X'0000004A'
MQIA_CF_LEVEL	70	X'00000046'
MQIA_CF_RECOVER	71	X'00000047'
MQIA_CHANNEL_AUTO_DEF	55	X'00000037'
MQIA_CHANNEL_AUTO_DEF_EVENT	56	X'00000038'
MQIA_CHANNEL_EVENT	73	X'00000049'
MQIA_CHECK_CLIENT_BINDING	258	X'00000102'
MQIA_CHECK_LOCAL_BINDING	257	X'00000101'
MQIA_CHINIT_ADAPTERS	101	X'00000065'
MQIA_CHINIT_CONTROL	119	X'00000077'
MQIA_CHINIT_DISPATCHERS	105	X'00000069'
MQIA_CHINIT_TRACE_AUTO_START	117	X'00000075'
MQIA_CHINIT_TRACE_TABLE_SIZE	118	X'00000076'
MQIA_CLUSTER_OBJECT_STATE	256	X'00000100'
MQIA_CLUSTER_PUB_ROUTE	255	X'000000FF'
MQIA_CLUSTER_Q_TYPE	59	X'0000003B'
MQIA_CLUSTER_WORKLOAD_LENGTH	58	X'0000003A'
MQIA_CLWL_MRU_CHANNELS	97	X'00000061'
MQIA_CLWL_Q_RANK	95	X'0000005F'
MQIA_CLWL_Q_PRIORITY	96	X'00000060'
MQIA_CLWL_USEQ	98	X'00000062'
MQIA_CMD_SERVER_AUTO	87	X'00000057'
MQIA_CMD_SERVER_CONTROL	120	X'00000078'
MQIA_CMD_SERVER_CONVERT_MSG	88	X'00000058'
MQIA_CMD_SERVER_DLQ_MSG	89	X'00000059'
MQIA_CODED_CHAR_SET_ID	2	X'00000002'
MQIA_COMM_EVENT	232	X'000000E8'
MQIA_COMMAND_EVENT	99	X'00000063'
MQIA_COMMAND_LEVEL	31	X'0000001F'
MQIA_CONFIGURATION_EVENT	51	X'00000033'

Tabela 184. Valores de constantes (continuação)

Nome	Valor decimal	Valor Hexadecimal
MQIA_CPI_LEVEL	27	X'0000001B'
MQIA_CURRENT_Q_DEPTH	3	X'00000003'
MQIA_DEF_BIND	61	X'0000003D'
MQIA_DEF_CLUSTER_XMIT_Q_TYPE	250	X'000000FA'
MQIA_DEF_INPUT_OPEN_OPTION	4	X'00000004'
MQIA_DEF_PERSISTENCE	5	X'00000005'
MQIA_DEF_PRIORITY	6	X'00000006'
MQIA_DEF_PUT_RESPONSE_TYPE	184	X'000000B8'
MQIA_DEF_READ_AHEAD	188	X'000000BC'
MQIA_DEFINITION_TYPE	7	X'00000007'
MQIA_DISPLAY_TYPE	262	X'00000106'
MQIA_DIST_LISTS	34	X'00000022'
MQIA_DNS_WLM	106	X'0000006A'
MQIA_DURABLE_SUB	175	X'000000AF'
MQIA_EXPIRY_INTERVAL	39	X'00000027'
MQIA_FIRST	1	X'00000001'
MQIA_GROUP_UR	221	X'000000DD'
MQIA_HARDEN_GET_BACKOUT	8	X'00000008'
MQIA_HIGH_Q_DEPTH	36	X'00000024'
MQIA_IGQ_PUT_AUTHORITY	65	X'00000041'
MQIA_INDEX_TYPE	57	X'00000039'
MQIA_INHIBIT_EVENT	48	X'00000030'
MQIA_INHIBIT_GET	9	X'00000009'
MQIA_INHIBIT_PUB	181	X'000000B5'
MQIA_INHIBIT_PUT	10	X'0000000A'
MQIA_INHIBIT_SUB	182	X'000000B6'
MQIA_INTRA_GROUP_queuing	64	X'00000040'
MQIA_IP_ADDRESS_VERSION	93	X'0000005D'
MQIA_KEY_REUSE_COUNT	267	X'0000010B'
MQIA_LAST	2000	X'000007D0'
MQIA_LAST_USED	267	X'0000010B'
MQIA_LDAP_AUTHORMD	263	X'00000107'
MQIA_LDAP_NESTGRP	264	X'00000108'
MQIA_LDAP_SECURE_COMM	261	X'00000105'
MQIA_LISTENER_PORT_NUMBER	85	X'00000055'
MQIA_LISTENER_TIMER	107	X'0000006B'
MQIA_LOGGER_EVENT	94	X'0000005E'
MQIA_LU62_CHANNELS	108	X'0000006C'
MQIA_LOCAL_EVENT	49	X'00000031'



Tabela 184. Valores de constantes (continuação)

Nome	Valor decimal	Valor Hexadecimal
MQIA_MSG_MARK_BROWSE_INTERVAL	68	X'00000044'
MQIA_MAX_CHANNELS	109	X'0000006D'
MQIA_MAX_CLIENTS	172	X'000000AC'
MQIA_MAX_GLOBAL_LOCKS	83	X'00000053'
MQIA_MAX_HANDLES	11	X'0000000B'
MQIA_MAX_LOCAL_LOCKS	84	X'00000054'
MQIA_MAX_MSG_LENGTH	13	X'0000000D'
MQIA_MAX_OPEN_Q	80	X'00000050'
MQIA_MAX_PRIORITY	14	X'0000000E'
MQIA_MAX_PROPERTIES_LENGTH	192	X'000000C0'
MQIA_MAX_Q_DEPTH	15	X'0000000F'
MQIA_MAX_Q_TRIGGERS	90	X'0000005A'
MQIA_MAX_RECOVERY_TASKS	171	X'000000AB'
MQIA_MAX_UNCOMMITTED_MSGS	33	X'00000021'
MQIA_MCAST_BRIDGE	233	X'000000E9'
MQIA_MONITOR_INTERVAL	81	X'00000051'
MQIA_MONITORING_AUTO_CLUSSDR	124	X'0000007C'
MQIA_MONITORING_CHANNEL	122	X'0000007A'
MQIA_MONITORING_Q	123	X'0000007B'
MQIA_MSG_DELIVERY_SEQUENCE	16	X'00000010'
MQIA_MSG_DEQ_COUNT	38	X'00000026'
MQIA_MSG_ENQ_COUNT	37	X'00000025'
MQIA_NAME_COUNT	19	X'00000013'
MQIA_NAMELIST_TYPE	72	X'00000048'
MQIA_NPM_CLASS	78	X'0000004E'
MQIA_NPM_DELIVERY	196	X'000000C4'
MQIA_OPEN_INPUT_COUNT	17	X'00000011'
MQIA_OPEN_OUTPUT_COUNT	18	X'00000012'
MQIA_OUTBOUND_PORT_MAX	140	X'0000008C'
MQIA_OUTBOUND_PORT_MIN	110	X'0000006E'
MQIA_PAGESET_ID	62	X'0000003E'
MQIA_PERFORMANCE_EVENT	53	X'00000035'
MQIA_PLATFORM	32	X'00000020'
MQIA_PM_DELIVERY	195	X'000000C3'
MQIA_PROPERTY_CONTROL	190	X'000000BE'
MQIA_PROT_POLICY_CAPABILITY	251	X'000000FB'
MQIA_PROXY_SUB	199	X'000000C7'
MQIA_PUB_COUNT	215	X'000000D7'
MQIA_PUB_SCOPE	219	X'000000DB'

Tabela 184. Valores de constantes (continuação)

Nome	Valor decimal	Valor Hexadecimal
MQIA_PUBSUB_CLUSTER	249	X'000000F9'
MQIA_PUBSUB_MAXMSG_RETRY_COUNT	206	X'000000CE'
MQIA_PUBSUB_MODE	187	X'000000BB'
MQIA_PUBSUB_NP_MSG	203	X'000000CB'
MQIA_PUBSUB_NP_RESP	205	X'000000CD'
MQIA_PUBSUB_SYNC_PT	207	X'000000CF'
MQIA_Q_DEPTH_HIGH_EVENT	43	X'0000002B'
MQIA_Q_DEPTH_HIGH_LIMIT	40	X'00000028'
MQIA_Q_DEPTH_LOW_EVENT	44	X'0000002C'
MQIA_Q_DEPTH_LOW_LIMIT	41	X'00000029'
MQIA_Q_DEPTH_MAX_EVENT	42	X'0000002A'
MQIA_Q_SERVICE_INTERVAL	54	X'00000036'
MQIA_Q_SERVICE_INTERVAL_EVENT	46	X'0000002E'
MQIA_Q_TYPE	20	X'00000014'
MQIA_Q_USERS	82	X'00000052'
MQIA_QMGR_CFCONLOS	245	X'000000F5'
MQIA_QMOPT_CONS_COMMS_MSGS	155	X'0000009B'
MQIA_QMOPT_CONS_CRITICAL_MSGS	154	X'0000009A'
MQIA_QMOPT_CONS_ERROR_MSGS	153	X'00000099'
MQIA_QMOPT_CONS_INFO_MSGS	151	X'00000097'
MQIA_QMOPT_CONS_REORG_MSGS	156	X'0000009C'
MQIA_QMOPT_CONS_SYSTEM_MSGS	157	X'0000009D'
MQIA_QMOPT_CONS_WARNING_MSGS	152	X'00000098'
MQIA_QMOPT_CSMT_ON_ERROR	150	X'00000096'
MQIA_QMOPT_INTERNAL_DUMP	170	X'000000AA'
MQIA_QMOPT_LOG_COMMS_MSGS	162	X'000000A2'
MQIA_QMOPT_LOG_CRITICAL_MSGS	161	X'000000A1'
MQIA_QMOPT_LOG_ERROR_MSGS	160	X'000000A0'
MQIA_QMOPT_LOG_INFO_MSGS	158	X'0000009E'
MQIA_QMOPT_LOG_REORG_MSGS	163	X'000000A3'
MQIA_QMOPT_LOG_SYSTEM_MSGS	164	X'000000A4'
MQIA_QMOPT_LOG_WARNING_MSGS	159	X'0000009F'
MQIA_QMOPT_TRACE_COMMS	166	X'000000A6'
MQIA_QMOPT_TRACE_CONVERSION	168	X'000000A8'
MQIA_QMOPT_TRACE_REORG	167	X'000000A7'
MQIA_QMOPT_TRACE_MQI_CALLS	165	X'000000A5'
MQIA_QMOPT_TRACE_SYSTEM	169	X'000000A9'
MQIA_QSG_DISP	63	X'0000003F'
MQIA_READ_AHEAD	189	X'000000BD'

Tabela 184. Valores de constantes (continuação)

Nome	Valor decimal	Valor Hexadecimal
MQIA_RECEIVE_TIMEOUT	111	X'0000006F'
MQIA_RECEIVE_TIMEOUT_MIN	113	X'00000071'
MQIA_RECEIVE_TIMEOUT_TYPE	112	X'00000070'
MQIA_REMOTE_EVENT	50	X'00000032'
MQIA_RETENTION_INTERVAL	21	X'00000015'
MQIA_REVERSE_DNS_LOOKUP	254	X'000000FE'
MQIA_SCOPE	45	X'0000002D'
MQIA_SECURITY_CASE	141	X'0000008D'
MQIA_SERVICE_CONTROL	139	X'0000008B'
MQIA_SERVICE_TYPE	121	X'00000079'
MQIA_SHAREABILITY	23	X'00000017'
MQIA_SHARED_Q_Q_MGR_NAME	77	X'0000004D'
MQIA_SSL_EVENT	75	X'0000004B'
MQIA_SSL_FIPS_REQUIRED	92	X'0000005C'
MQIA_SSL_RESET_COUNT	76	X'0000004C'
MQIA_SSL_TASKS	69	X'00000045'
MQIA_START_STOP_EVENT	52	X'00000034'
MQIA_STATISTICS_CHANNEL	129	X'00000081'
MQIA_STATISTICS_AUTO_CLUSSDR	130	X'00000082'
MQIA_STATISTICS_INTERVAL	131	X'00000083'
MQIA_STATISTICS_MQI	127	X'0000007F'
MQIA_STATISTICS_Q	128	X'00000080'
MQIA_SUB_COUNT	204	X'000000CC'
MQIA_SUB_SCOPE	218	X'000000DA'
MQIA_SYNCPOINT	30	X'0000001E'
MQIA_TCP_CHANNELS	114	X'00000072'
MQIA_TCP_KEEP_ALIVE	115	X'00000073'
MQIA_TCP_STACK_TYPE	116	X'00000074'
MQIA_TIME_SINCE_RESET	35	X'00000023'
MQIA_TOPIC_DEF_PERSISTENCE	185	X'000000B9'
MQIA_TOPIC_NODE_COUNT	253	X'000000FD'
MQIA_TOPIC_TYPE	208	X'000000D0'
MQIA_TRACE_ROUTE_RECORDING	137	X'00000089'
MQIA_TREE_LIFE_TIME	183	X'000000B7'
MQIA_TRIGGER_CONTROL	24	X'00000018'
MQIA_TRIGGER_DEPTH	29	X'0000001D'
MQIA_TRIGGER_INTERVAL	25	X'00000019'
MQIA_TRIGGER_MSG_PRIORITY	26	X'0000001A'
MQIA_TRIGGER_TYPE	28	X'0000001C'

Tabela 184. Valores de constantes (continuação)

Nome	Valor decimal	Valor Hexadecimal
MQIA_TRIGGER_RESTART	91	X'0000005B'
MQIA_USAGE	12	X'0000000C'
MQIA_USE_DEAD_LETTER_Q	234	X'000000EA'
MQIA_USER_LIST	2000	X'000007D0'
MQIA_WILDCARD_OPERATION	216	X'000000D8'
MQIA_XR_CAPABILITY	243	X'000000F3'

## MQIACF\_\* (Tipos de Parâmetros de Número Inteiro do formato de comando)

Tabela 185. Valores de constantes

Nome	Valor decimal	Valor Hexadecimal
MQIACF_FIRST	1001	X'000003E9'
MQIACF_Q_MGR_ATTRS	1001	X'000003E9'
MQIACF_Q_ATTRS	1002	X'000003EA'
MQIACF_PROCESS_ATTRS	1003	X'000003EB'
MQIACF_NAMELIST_ATTRS	1004	X'000003EC'
MQIACF_FORCE	1005	X'000003ED'
MQIACF_REPLACE	1006	X'000003EE'
MQIACF_PURGE	1007	X'000003EF'
MQIACF QUIESCE	1008	X'000003F0'
MQIACF_MODE	1008	X'000003F0'
MQIACF_ALL	1009	X'000003F1'
MQIACF_EVENT_APPL_TYPE	1010	X'000003F2'
MQIACF_EVENT_ORIGIN	1011	X'000003F3'
MQIACF_PARAMETER_ID	1012	X'000003F4'
MQIACF_ERROR_ID	1013	X'000003F5'
MQIACF_ERROR_IDENTIFIER	1013	X'000003F5'
MQIACF_SELECTOR	1014	X'000003F6'
MQIACF_CHANNEL_ATTRS	1015	X'000003F7'
MQIACF_OBJECT_TYPE	1016	X'000003F8'
MQIACF_ESCAPE_TYPE	1017	X'000003F9'
MQIACF_ERROR_OFFSET	1018	X'000003FA'
MQIACF_AUTH_INFO_ATTRS	1019	X'000003FB'
MQIACF_REASON_QUALIFIER	1020	X'000003FC'
MQIACF_COMMAND	1021	X'000003FD'
MQIACF_OPEN_OPTIONS	1022	X'000003FE'
MQIACF_OPEN_TYPE	1023	X'000003FF'
MQIACF_PROCESS_ID	1024	X'00000400'
MQIACF_THREAD_ID	1025	X'00000401'
MQIACF_Q_STATUS_ATTRS	1026	X'00000402'

Tabela 185. Valores de constantes (continuação)

Nome	Valor decimal	Valor Hexadecimal
MQIACF_UNCOMMITTED_MSGS	1027	X'00000403'
MQIACF_HANDLE_STATE	1028	X'00000404'
MQIACF_AUX_ERROR_DATA_INT_1	1070	X'0000042E'
MQIACF_AUX_ERROR_DATA_INT_2	1071	X'0000042F'
MQIACF_CONV_REASON_CODE	1072	X'00000430'
MQIACF_BRIDGE_TYPE	1073	X'00000431'
MQIACF_INQUIRY	1074	X'00000432'
MQIACF_WAIT_INTERVAL	1075	X'00000433'
MQIACF_OPTIONS	1076	X'00000434'
MQIACF_BROKER_OPTIONS	1077	X'00000435'
MQIACF_REFRESH_TYPE	1078	X'00000436'
MQIACF_SEQUENCE_NUMBER	1079	X'00000437'
MQIACF_INTEGER_DATA	1080	X'00000438'
MQIACF_REGISTRATION_OPTIONS	1081	X'00000439'
MQIACF_PUBLICATION_OPTIONS	1082	X'0000043A'
MQIACF_CLUSTER_INFO	1083	X'0000043B'
MQIACF_Q_MGR_DEFINITION_TYPE	1084	X'0000043C'
MQIACF_Q_MGR_TYPE	1085	X'0000043D'
MQIACF_ACTION	1086	X'0000043E'
MQIACF_SUSPEND	1087	X'0000043F'
MQIACF_BROKER_COUNT	1088	X'00000440'
MQIACF_APPL_COUNT	1089	X'00000441'
MQIACF_ANONYMOUS_COUNT	1090	X'00000442'
MQIACF_REG_REG_OPTIONS	1091	X'00000443'
MQIACF_DELETE_OPTIONS	1092	X'00000444'
MQIACF_CLUSTER_Q_MGR_ATTRS	1093	X'00000445'
MQIACF_REFRESH_INTERVAL	1094	X'00000446'
MQIACF_REFRESH_REPOSITORY	1095	X'00000447'
MQIACF_REMOVE_QUEUES	1096	X'00000448'
MQIACF_OPEN_INPUT_TYPE	1098	X'0000044A'
MQIACF_OPEN_OUTPUT	1099	X'0000044B'
MQIACF_OPEN_SET	1100	X'0000044C'
MQIACF_OPEN_INQUIRE	1101	X'0000044D'
MQIACF_OPEN_BROWSE	1102	X'0000044E'
MQIACF_Q_STATUS_TYPE	1103	X'0000044F'
MQIACF_Q_HANDLE	1104	X'00000450'
MQIACF_Q_STATUS	1105	X'00000451'
MQIACF_SECURITY_TYPE	1106	X'00000452'
MQIACF_CONNECTION_ATTRS	1107	X'00000453'

Tabela 185. Valores de constantes (continuação)

Nome	Valor decimal	Valor Hexadecimal
MQIACF_CONNECT_OPTIONS	1108	X'00000454'
MQIACF_CONN_INFO_TYPE	1110	X'00000456'
MQIACF_CONN_INFO_CONN	1111	X'00000457'
MQIACF_CONN_INFO_HANDLE	1112	X'00000458'
MQIACF_CONN_INFO_ALL	1113	X'00000459'
MQIACF_AUTH_PROFILE_ATTRS	1114	X'0000045A'
MQIACF_AUTHORIZATION_LIST	1115	X'0000045B'
MQIACF_AUTH_ADD_AUTHS	1116	X'0000045C'
MQIACF_AUTH_REMOVE_AUTHS	1117	X'0000045D'
MQIACF_ENTITY_TYPE	1118	X'0000045E'
MQIACF_COMMAND_INFO	1120	X'00000460'
MQIACF_CMDScope_Q_MGR_COUNT	1121	X'00000461'
MQIACF_Q_MGR_SYSTEM	1122	X'00000462'
MQIACF_Q_MGR_EVENT	1123	X'00000463'
MQIACF_Q_MGR_DQM	1124	X'00000464'
MQIACF_Q_MGR_CLUSTER	1125	X'00000465'
MQIACF_QSG_DISPS	1126	X'00000466'
MQIACF_UOW_STATE	1128	X'00000468'
MQIACF_SECURITY_ITEM	1129	X'00000469'
MQIACF_CF_STRUC_STATUS	1130	X'0000046A'
MQIACF_UOW_TYPE	1132	X'0000046C'
MQIACF_CF_STRUC_ATTRS	1133	X'0000046D'
MQIACF_EXCLUDE_INTERVAL	1134	X'0000046E'
MQIACF_CF_STATUS_TYPE	1135	X'0000046F'
MQIACF_CF_STATUS_SUMMARY	1136	X'00000470'
MQIACF_CF_STATUS_CONNECT	1137	X'00000471'
MQIACF_CF_STATUS_BACKUP	1138	X'00000472'
MQIACF_CF_STRUC_TYPE	1139	X'00000473'
MQIACF_CF_STRUC_SIZE_MAX	1140	X'00000474'
MQIACF_CF_STRUC_SIZE_USED	1141	X'00000475'
MQIACF_CF_STRUC_ENTRIES_MAX	1142	X'00000476'
MQIACF_CF_STRUC_ENTRIES_USED	1143	X'00000477'
MQIACF_CF_STRUC_BACKUP_SIZE	1144	X'00000478'
MQIACF_MOVE_TYPE	1145	X'00000479'
MQIACF_MOVE_TYPE_MOVE	1146	X'0000047A'
MQIACF_MOVE_TYPE_ADD	1147	X'0000047B'
MQIACF_Q_MGR_NUMBER	1148	X'0000047C'
MQIACF_Q_MGR_STATUS	1149	X'0000047D'
MQIACF_DB2_CONN_STATUS	1150	X'0000047E'

Tabela 185. Valores de constantes (continuação)

Nome	Valor decimal	Valor Hexadecimal
MQIACF_SECURITY_ATTRS	1151	X'0000047F'
MQIACF_SECURITY_TIMEOUT	1152	X'00000480'
MQIACF_SECURITY_INTERVAL	1153	X'00000481'
MQIACF_SECURITY_SWITCH	1154	X'00000482'
MQIACF_SECURITY_SETTING	1155	X'00000483'
MQIACF_STORAGE_CLASS_ATTRS	1156	X'00000484'
MQIACF_USAGE_TYPE	1157	X'00000485'
MQIACF_BUFFER_POOL_ID	1158	X'00000486'
MQIACF_USAGE_TOTAL_PAGES	1159	X'00000487'
MQIACF_USAGE_UNUSED_PAGES	1160	X'00000488'
MQIACF_USAGE_PERSIST_PAGES	1161	X'00000489'
MQIACF_USAGE_NONPERSIST_PAGES	1162	X'0000048A'
MQIACF_USAGE_RESTART_EXTENTS	1163	X'0000048B'
MQIACF_USAGE_EXPAND_COUNT	1164	X'0000048C'
MQIACF_PAGESET_STATUS	1165	X'0000048D'
MQIACF_USAGE_TOTAL_BUFFERS	1166	X'0000048E'
MQIACF_USAGE_DATA_SET_TYPE	1167	X'0000048F'
MQIACF_USAGE_PAGESET	1168	X'00000490'
MQIACF_USAGE_DATA_SET	1169	X'00000491'
MQIACF_USAGE_BUFFER_POOL	1170	X'00000492'
MQIACF_MOVE_COUNT	1171	X'00000493'
MQIACF_EXPIRY_Q_COUNT	1172	X'00000494'
MQIACF_CONFIGURATION_OBJECTS	1173	X'00000495'
MQIACF_CONFIGURATION_EVENTS	1174	X'00000496'
MQIACF_SYSP_TYPE	1175	X'00000497'
MQIACF_SYSP_DEALLOC_INTERVAL	1176	X'00000498'
MQIACF_SYSP_MAX_ARCHIVE	1177	X'00000499'
MQIACF_SYSP_MAX_READ_TAPES	1178	X'0000049A'
MQIACF_SYSP_IN_BUFFER_SIZE	1179	X'0000049B'
MQIACF_SYSP_OUT_BUFFER_SIZE	1180	X'0000049C'
MQIACF_SYSP_OUT_BUFFER_COUNT	1181	X'0000049D'
MQIACF_SYSP_ARCHIVE	1182	X'0000049E'
MQIACF_SYSP_DUAL_ACTIVE	1183	X'0000049F'
MQIACF_SYSP_DUAL_ARCHIVE	1184	X'000004A0'
MQIACF_SYSP_DUAL_BSDS	1185	X'000004A1'
MQIACF_SYSP_MAX_CONNS	1186	X'000004A2'
MQIACF_SYSP_MAX_CONNS_FORE	1187	X'000004A3'
MQIACF_SYSP_MAX_CONNS_BACK	1188	X'000004A4'
MQIACF_SYSP_EXIT_INTERVAL	1189	X'000004A5'

Tabela 185. Valores de constantes (continuação)

Nome	Valor decimal	Valor Hexadecimal
MQIACF_SYSP_EXIT_TASKS	1190	X'000004A6'
MQIACF_SYSP_CHKPOINT_COUNT	1191	X'000004A7'
MQIACF_SYSP_OTMA_INTERVAL	1192	X'000004A8'
MQIACF_SYSP_Q_INDEX_DEFER	1193	X'000004A9'
MQIACF_SYSP_DB2_TASKS	1194	X'000004AA'
MQIACF_SYSP_RESLEVEL_AUDIT	1195	X'000004AB'
MQIACF_SYSP_ROUTING_CODE	1196	X'000004AC'
MQIACF_SYSP_SMF_ACCOUNTING	1197	X'000004AD'
MQIACF_SYSP_SMF_STATS	1198	X'000004AE'
MQIACF_SYSP_SMF_INTERVAL	1199	X'000004AF'
MQIACF_SYSP_TRACE_CLASS	1200	X'000004B0'
MQIACF_SYSP_TRACE_SIZE	1201	X'000004B1'
MQIACF_SYSP_WLM_INTERVAL	1202	X'000004B2'
MQIACF_SYSP_ALLOC_UNIT	1203	X'000004B3'
MQIACF_SYSP_ARCHIVE_RETAIN	1204	X'000004B4'
MQIACF_SYSP_ARCHIVE_WTOR	1205	X'000004B5'
MQIACF_SYSP_BLOCK_SIZE	1206	X'000004B6'
MQIACF_SYSP_CATALOG	1207	X'000004B7'
MQIACF_SYSP_COMPACT	1208	X'000004B8'
MQIACF_SYSP_ALLOC_PRIMARY	1209	X'000004B9'
MQIACF_SYSP_ALLOC_SECONDARY	1210	X'000004BA'
MQIACF_SYSP_PROTECT	1211	X'000004BB'
MQIACF_SYSP_QUIESCE_INTERVAL	1212	X'000004BC'
MQIACF_SYSP_TIMESTAMP	1213	X'000004BD'
MQIACF_SYSP_UNIT_ADDRESS	1214	X'000004BE'
MQIACF_SYSP_UNIT_STATUS	1215	X'000004BF'
MQIACF_SYSP_LOG_COPY	1216	X'000004C0'
MQIACF_SYSP_LOG_USED	1217	X'000004C1'
MQIACF_SYSP_LOG_SUSPEND	1218	X'000004C2'
MQIACF_SYSP_OFFLOAD_STATUS	1219	X'000004C3'
MQIACF_SYSP_TOTAL_LOGS	1220	X'000004C4'
MQIACF_SYSP_FULL_LOGS	1221	X'000004C5'
MQIACF_LISTENER_ATTRS	1222	X'000004C6'
MQIACF_LISTENER_STATUS_ATTRS	1223	X'000004C7'
MQIACF_SERVICE_ATTRS	1224	X'000004C8'
MQIACF_SERVICE_STATUS_ATTRS	1225	X'000004C9'
MQIACF_Q_TIME_INDICATOR	1226	X'000004CA'
MQIACF_OLDEST_MSG_AGE	1227	X'000004CB'
MQIACF_AUTH_OPTIONS	1228	X'000004CC'



Tabela 185. Valores de constantes (continuação)

Nome	Valor decimal	Valor Hexadecimal
MQIACF_Q_MGR_STATUS_ATTRS	1229	X'000004CD'
MQIACF_CONNECTION_COUNT	1230	X'000004CE'
MQIACF_Q_MGR_FACILITY	1231	X'000004CF'
MQIACF_CHINIT_STATUS	1232	X'000004D0'
MQIACF_CMD_SERVER_STATUS	1233	X'000004D1'
MQIACF_ROUTE_DETAIL	1234	X'000004D2'
MQIACF_RECORDED_ACTIVITIES	1235	X'000004D3'
MQIACF_MAX_ACTIVIDADES	1236	X'000004D4'
MQIACF_DISCONTINUITY_COUNT	1237	X'000004D5'
MQIACF_ROUTE_ACUMULAÇÃO	1238	X'000004D6'
MQIACF_ROUTE_DELIVERY	1239	X'000004D7'
MQIACF_OPERATION_TYPE	1240	X'000004D8'
MQIACF_BACKOUT_COUNT	1241	X'000004D9'
MQIACF_COMP_CODE	1242	X'000004DA'
MQIACF_ENCODING	1243	X'000004DB'
MQIACF_EXPIRY	1244	X'000004DC'
MQIACF_FEEDBACK	1245	X'000004DD'
MQIACF_MSG_FLAGS	1247	X'000004DF'
MQIACF_MSG_LENGTH	1248	X'000004E0'
MQIACF_MSG_TYPE	1249	X'000004E1'
MQIACF_OFFSET	1250	X'000004E2'
MQIACF_ORIGINAL_LENGTH	1251	X'000004E3'
MQIACF_PERSISTENCE	1252	X'000004E4'
MQIACF_PRIORITY	1253	X'000004E5'
MQIACF_REASON_CODE	1254	X'000004E6'
MQIACF_REPORT	1255	X'000004E7'
MQIACF_VERSION	1256	X'000004E8'
MQIACF_UNRECORDED_ACTIVITIES	1257	X'000004E9'
MQIACF_MONITORING	1258	X'000004EA'
MQIACF_ROUTE_FORWARDING	1259	X'000004EB'
MQIACF_SERVICE_STATUS	1260	X'000004EC'
MQIACF_Q_TYPES	1261	X'000004ED'
MQIACF_USER_ID_SUPPORT	1262	X'000004EE'
MQIACF_INTERFACE_VERSION	1263	X'000004EF'
MQIACF_AUTH_SERVICE_ATTRS	1264	X'000004F0'
MQIACF_USAGE_EXPAND_TYPE	1265	X'000004F1'
MQIACF_SYSP_CLUSTER_CACHE	1266	X'000004F2'
MQIACF_SYSP_DB2_BLOB_TASKS	1267	X'000004F3'
MQIACF_SYSP_WLM_INT_UNITS	1268	X'000004F4'

Tabela 185. Valores de constantes (continuação)

Nome	Valor decimal	Valor Hexadecimal
MQIACF_TOPIC_ATTRS	1269	X'000004F5'
MQIACF_PUBSUB_PROPERTIES	1271	X'000004F7'
MQIACF_DESTINATION_CLASS	1273	X'000004F9'
MQIACF_DURABLE_SUBSCRIPTION	1274	X'000004FA'
MQIACF_SUBSCRIPTION_SCOPE	1275	X'000004FB'
MQIACF_VARIABLE_USER_ID	1277	X'000004FD'
MQIACF_REQUEST_ONLY	1280	X'00000500'
MQIACF_PUB_PRIORITY	1283	X'00000503'
MQIACF_SUB_ATTRS	1287	X'00000507'
MQIACF_WILDCARD_SCHEMA	1288	X'00000508'
MQIACF_SUB_TYPE	1289	X'00000509'
MQIACF_MESSAGE_COUNT	1290	X'0000050A'
MQIACF_Q_MGR_PUBSUB	1291	X'0000050B'
MQIACF_Q_MGR_VERSION	1292	X'0000050C'
MQIACF_SUB_STATUS_ATTRS	1294	X'0000050E'
MQIACF_TOPIC_STATUS	1295	X'0000050F'
MQIACF_TOPIC_SUB	1296	X'00000510'
MQIACF_TOPIC_PUB	1297	X'00000511'
MQIACF_RETAINED_PUBLICATION	1300	X'00000514'
MQIACF_TOPIC_STATUS_ATTRS	1301	X'00000515'
MQIACF_TOPIC_STATUS_TYPE	1302	X'00000516'
MQIACF_SUB_OPTIONS	1303	X'00000517'
MQIACF_PUBLISH_COUNT	1304	X'00000518'
MQIACF_CLEAR_TYPE	1305	X'00000519'
MQIACF_CLEAR_SCOPE	1306	X'0000051A'
MQIACF_SUB_LEVEL	1307	X'0000051B'
MQIACF_ASYNC_STATE	1308	X'0000051C'
MQIACF_SUB_SUMMARY	1309	X'0000051D'
MQIACF_OBSOLETE_MSGS	1310	X'0000051E'
MQIACF_PUBSUB_STATUS	1311	X'0000051F'
MQIACF_PS_STATUS_TYPE	1314	X'00000522'
MQIACF_PUBSUB_STATUS_ATTRS	1318	X'00000526'
MQIACF_SELECTOR_TYPE	1321	X'00000529'
MQIACF_MCAST_REL_INDICATOR	1351	X'00000547'
MQIACF_CHLAUTH_TYPE	1352	X'00000548'
MQXR_DIAGNOSTICS_TYPE	1354	X'0000054A'
MQIACF_CHLAUTH_ATTRS	1355	X'0000054B'
MQIACF_OPERATION_ID	1356	X'0000054C'
MQIACF_API_CALLER_TYPE	1357	X'0000054D'


Tabela 185. Valores de constantes (continuação)

Nome	Valor decimal	Valor Hexadecimal
MQIACF_API_ENVIRONMENT	1358	X'0000054E'
MQIACF_TRACE_DETAIL	1359	X'0000054F'
MQIACF_HOBJ	1360	X'00000550'
MQIACF_CALL_TYPE	1361	X'00000551'
MQIACF_MQCB_OPERATION	1362	X'00000552'
MQIACF_MQCB_TYPE	1363	X'00000553'
MQIACF_MQCB_OPTIONS	1364	X'00000554'
MQIACF_CLOSE_OPTIONS	1365	X'00000555'
MQIACF_CTL_OPERATION	1366	X'00000556'
MQIACF_GET_OPTIONS	1367	X'00000557'
MQIACF_RECS_PRESENT	1368	X'00000558'
MQIACF_KNOWN_DEST_COUNT	1369	X'00000559'
MQIACF_UNKNOWN_DEST_COUNT	1370	X'0000055A'
MQIACF_INVALID_DEST_COUNT	1371	X'0000055B'
MQIACF_RESOLVED_TYPE	1372	X'0000055C'
MQIACF_PUT_OPTIONS	1373	X'0000055D'
MQIACF_BUFFER_LENGTH	1374	X'0000055E'
MQIACF_TRACE_DATA_LENGTH	1375	X'0000055F'
MQIACF_SMDS_EXPANDST	1376	X'00000560'
MQIACF_STRUC_LENGTH	1377	X'00000561'
MQIACF_ITEM_COUNT	1378	X'00000562'
MQIACF_EXPIRY_TIME	1379	X'00000563'
MQIACF_CONNECT_TIME	1380	X'00000564'
MQIACF_DISCONNECT_TIME	1381	X'00000565'
MQIACF_HSUB	1382	X'00000566'
MQIACF_SUBRQ_OPTIONS	1383	X'00000567'
MQIACF_XA_RMID	1384	X'00000568'
MQIACF_XA_FLAGS	1385	X'00000569'
MQIACF_XA_RETCODE	1386	X'0000056A'
MQIACF_XA_HANDLE	1387	X'0000056B'
MQIACF_XA_RETVAL	1388	X'0000056C'
MQIACF_STATUS_TYPE	1389	X'0000056D'
MQIACF_XA_COUNT	1390	X'0000056E'
MQIACF_SELECTOR_COUNT	1391	X'0000056F'
MQIACF_SELECTORS	1392	X'00000570'
MQIACF_INTATTR_COUNT	1393	X'00000571'
MQIACF_INTATTRS	1394	X'00000572'
MQIACF_SUBRQ_ACTION	1395	X'00000573'
MQIACF_NUM_PUBS	1396	X'00000574'

Tabela 185. Valores de constantes (continuação)

Nome	Valor decimal	Valor Hexadecimal
MQIACF_POINTER_SIZE	1397	X'00000575'
MQIACF_REMOVE_AUTHREC	1398	X'00000576'
MQIACF_XR_ATTRS	1399	X'00000577'
MQIACF_APPL_FUNCTION_TYPE	1400	X'00000578'
MQIACF_AMQP_ATTRS	1401	X'00000579'
MQIACF_EXPORT_TYPE	1402	X'0000057A'
MQIACF_EXPORT_ATTRS	1403	X'0000057B'
MQIACF_SYSTEM_OBJECTS	1404	X'0000057C'
MQIACF_CONNECTION_SWAP	1405	X'0000057D'
MQIACF_AMQP_DIAGNOSTICS_TYPE	1406	X'0000057E'
MQIACF_BUFFER_POOL_LOCATION	1408	X'00000580'
MQIACF_LDAP_CONNECTION_STATUS	1409	X'00000581'
MQIACF_SYSP_MAX_ACE_POOL	1410	X'00000582'
MQIACF_PAGECLAS	1411	X'00000583'
MQIACF_AUTH_REC_TYPE	1412	X'00000584'
MQIACF_SYSP_MAX_CONC_OFFLOADS	1413	X'00000585'
MQIACF_SYSP_ZHYPERWRITE	1414	X'00000586'
MQIACF_Q_MGR_STATUS_LOG	1415	X'00000587'
MQIACF_ARCHIVE_LOG_SIZE	1416	X'00000588'
MQIACF_MEDIA_LOG_SIZE	1417	X'00000589'
MQIACF_RESTART_LOG_SIZE	1418	X'0000058A'
MQIACF_REUSABLE_LOG_SIZE	1419	X'0000058B'
MQIACF_LOG_IN_USE	1420	X'0000058C'
MQIACF_LOG_UTILIZAÇÃO	1421	X'0000058D'
MQIACF_IGNORE_STATE	1423	X'0000058F'
MQIACF_MOVABLE_APPL_COUNT	1424	X'00000590'
MQIACF_APPL_INFO_ATTRS	1425	X'00000591'
MQIACF_APPL_MOVABLE	1426	X'00000592'
MQIACF_REMOTE_QMGR_ACTIVE	1427	X'00000593'
MQIACF_APPL_INFO_TYPE	1428	X'00000594'
MQIACF_APPL_INFO_APPL	1429	X'00000595'
MQIACF_APPL_INFO_QMGR	1430	X'00000596'
MQIACF_APPL_INFO_LOCAL	1431	X'00000597'
MQIACF_APPL_IMMOVABLE_COUNT	1432	X'00000598'
MQIACF_BALANCED	1433	X'00000599'
MQIACF_BALSTATE	1434	X'0000059A'
MQIACF_APPL_IMMOVABLE_REASON	1435	X'0000059B'
MQIACF_DS_ENCRYPTED	1436	X'0000059C'
MQIACF_CUR_Q_FILE_SIZE	1437	X'0000059D'

Tabela 185. Valores de constantes (continuação)

Nome	Valor decimal	Valor Hexadecimal
MQIACF_CUR_MAX_FILE_SIZE	1438	X'0000059E'
MQIACF_BALANCING_TYPE	1439	X'0000059F'
MQIACF_BALANCING_OPTIONS	1440	X'000005A0'
MQIACF_BALANCING_TIMEOUT	1441	X'000005A1'
MQIACF_SYSP_SMF_STAT_TIME_SECS	1442	X'000005A2'
MQIACF_SYSP_SMF_ACCT_TIME_MINS	1443	X'000005A3'
MQIACF_SYSP_SMF_ACCT_TIME_SECS	1444	X'000005A4'
 MQIACF_LAST_USED	1444	X'000005A4'

## MQIACH\_\* (Tipos de Canal de Número Inteiro do formato de comando)

Tabela 186. Valores de constantes

Nome	Valor decimal	Valor Hexadecimal
MQIACH_FIRST	1501	X'000005DD'
MQIACH_XMIT_PROTOCOL_TYPE	1501	X'000005DD'
MQIACH_BATCH_SIZE	1502	X'000005DE'
MQIACH_DISC_INTERVAL	1503	X'000005DF'
MQIACH_SHORT_TIMER	1504	X'000005E0'
MQIACH_SHORT_RETRY	1505	X'000005E1'
MQIACH_LONG_TIMER	1506	X'000005E2'
MQIACH_LONG_RETRY	1507	X'000005E3'
MQIACH_PUT_AUTHORITY	1508	X'000005E4'
MQIACH_SEQUENCE_NUMBER_WRAP	1509	X'000005E5'
MQIACH_MAX_MSG_LENGTH	1510	X'000005E6'
MQIACH_CHANNEL_TYPE	1511	X'000005E7'
MQIACH_DATA_COUNT	1512	X'000005E8'
MQIACH_NAME_COUNT	1513	X'000005E9'
MQIACH_MSG_SEQUENCE_NUMBER	1514	X'000005EA'
MQIACH_DATA_CONVERSION	1515	X'000005EB'
MQIACH_IN_DOUBT	1516	X'000005EC'
MQIACH_MCA_TYPE	1517	X'000005ED'
MQIACH_SESSION_COUNT	1518	X'000005EE'
MQIACH_ADAPTER	1519	X'000005EF'
MQIACH_COMMAND_COUNT	1520	X'000005F0'
MQIACH_SOCKET	1521	X'000005F1'
MQIACH_PORT	1522	X'000005F2'
MQIACH_CHANNEL_INSTANCE_TYPE	1523	X'000005F3'
MQIACH_CHANNEL_INSTANCE_ATTRS	1524	X'000005F4'
MQIACH_CHANNEL_ERROR_DATA	1525	X'000005F5'
MQIACH_CHANNEL_TABLE	1526	X'000005F6'

Tabela 186. Valores de constantes (continuação)

Nome	Valor decimal	Valor Hexadecimal
MQIACH_CHANNEL_STATUS	1527	X'000005F7'
MQIACH_INDOUBT_STATUS	1528	X'000005F8'
MQIACH_LAST_SEQ_NUMBER	1529	X'000005F9'
MQIACH_LAST_SEQUENCE_NUMBER	1529	X'000005F9'
MQIACH_CURRENT_MSGS	1531	X'000005FB'
MQIACH_CURRENT_SEQ_NUMBER	1532	X'000005FC'
MQIACH_CURRENT_SEQUENCE_NUMBER	1532	X'000005FC'
MQIACH_SSL_RETURN_CODE	1533	X'000005FD'
MQIACH_MSGS	1534	X'000005FE'
MQIACH_BYTES_SENT	1535	X'000005FF'
MQIACH_BYTES_RCVD	1536	X'00000600'
MQIACH_BYTES_RECEIVED	1536	X'00000600'
MQIACH_BATCHES	1537	X'00000601'
MQIACH_BUFFERS_SENT	1538	X'00000602'
MQIACH_BUFFERS_RCVD	1539	X'00000603'
MQIACH_BUFFERS_RECEIVED	1539	X'00000603'
MQIACH_LONG_RETRIES_LEFT	1540	X'00000604'
MQIACH_SHORT_RETRIES_LEFT	1541	X'00000605'
MQIACH_MCA_STATUS	1542	X'00000606'
MQIACH_STOP_REQUESTED	1543	X'00000607'
MQIACH_MR_COUNT	1544	X'00000608'
MQIACH_MR_INTERVAL	1545	X'00000609'
MQIACH_NPM_SPEED	1562	X'0000061A'
MQIACH_HB_INTERVAL	1563	X'0000061B'
MQIACH_BATCH_INTERVAL	1564	X'0000061C'
MQIACH_NETWORK_PRIORITY	1565	X'0000061D'
MQIACH_KEEP_ALIVE_INTERVAL	1566	X'0000061E'
MQIACH_BATCH_HB	1567	X'0000061F'
MQIACH_SSL_CLIENT_AUTH	1568	X'00000620'
MQIACH_ALLOC_RETRY	1570	X'00000622'
MQIACH_ALLOC_FAST_TIMER	1571	X'00000623'
MQIACH_ALLOC_SLOW_TIMER	1572	X'00000624'
MQIACH_DISC_RETRY	1573	X'00000625'
MQIACH_PORT_NUMBER	1574	X'00000626'
MQIACH_HDR_COMPRESSION	1575	X'00000627'
MQIACH_MSG_COMPRESSION	1576	X'00000628'
MQIACH_CLWL_CHANNEL_RANK	1577	X'00000629'
MQIACH_CLWL_CHANNEL_PRIORITY	1578	X'0000062A'
MQIACH_CLWL_CHANNEL_WEIGHT	1579	X'0000062B'

Tabela 186. Valores de constantes (continuação)

Nome	Valor decimal	Valor Hexadecimal
MQIACH_CHANNEL_DISP	1580	X'0000062C'
MQIACH_INBOUND_DISP	1581	X'0000062D'
MQIACH_CHANNEL_TYPES	1582	X'0000062E'
MQIACH_ADAPS_STARTED	1583	X'0000062F'
MQIACH_ADAPS_MAX	1584	X'00000630'
MQIACH_DISPS_STARTED	1585	X'00000631'
MQIACH_DISPS_MAX	1586	X'00000632'
MQIACH_SSLTASKS_STARTED	1587	X'00000633'
MQIACH_SSLTASKS_MAX	1588	X'00000634'
MQIACH_CURRENT_CHL	1589	X'00000635'
MQIACH_CURRENT_CHL_MAX	1590	X'00000636'
MQIACH_CURRENT_CHL_TCP	1591	X'00000637'
MQIACH_CURRENT_CHL_LU62	1592	X'00000638'
MQIACH_ACTIVE_CHL	1593	X'00000639'
MQIACH_ACTIVE_CHL_MAX	1594	X'0000063A'
MQIACH_ACTIVE_CHL_PAUSED	1595	X'0000063B'
MQIACH_ACTIVE_CHL_STARTED	1596	X'0000063C'
MQIACH_ACTIVE_CHL_STOPPED	1597	X'0000063D'
MQIACH_ACTIVE_CHL_RETRY	1598	X'0000063E'
MQIACH_LISTENER_STATUS	1599	X'0000063F'
MQIACH_SHARED_CHL_RESTART	1600	X'00000640'
MQIACH_LISTENER_CONTROL	1601	X'00000641'
MQIACH_BACKLOG	1602	X'00000642'
MQIACH_XMITQ_TIME_INDICATOR	1604	X'00000644'
MQIACH_NETWORK_TIME_INDICATOR	1605	X'00000645'
MQIACH_EXIT_TIME_INDICATOR	1606	X'00000646'
MQIACH_BATCH_SIZE_INDICATOR	1607	X'00000647'
MQIACH_XMITQ_MSGS_AVAILABLE	1608	X'00000648'
MQIACH_CHANNEL_SUBSTATE	1609	X'00000649'
MQIACH_SSL_KEY_RESETS	1610	X'0000064A'
MQIACH_COMPRESSION_RATE	1611	X'0000064B'
MQIACH_COMPRESSION_TIME	1612	X'0000064C'
MQIACH_MAX_XMIT_SIZE	1613	X'0000064D'
MQIACH_DEF_CHANNEL_DISP	1614	X'0000064E'
MQIACH_SHARING_CONVERSATIONS	1615	X'0000064F'
MQIACH_MAX_SHARING_CONVS	1616	X'00000650'
MQIACH_CURRENT_SHARING_CONVS	1617	X'00000651'
MQIACH_MAX_INSTANCES	1618	X'00000652'
MQIACH_MAX_INSTS_PER_CLIENT	1619	X'00000653'

Tabela 186. Valores de constantes (continuação)

Nome	Valor decimal	Valor Hexadecimal
MQIACH_CLIENT_CHANNEL_WEIGHT	1620	X'00000654'
MQIACH_CONNECTION_AFFINITY	1621	X'00000655'
MQIACH_AUTH_INFO_TYPES	1622	X'00000656'
MQIACH_RESET_REQUESTED	1623	X'00000657'
MQIACH_BATCH_DATA_LIMIT	1624	X'00000658'
MQIACH_MSG_HISTORY	1625	X'00000659'
MQIACH_MULTICAST_PROPERTIES	1626	X'0000065A'
MQIACH_NEW_SUBSCRIBER_HISTORY	1627	X'0000065B'
MQIACH_MC_HB_INTERVAL	1628	X'0000065C'
MQIACH_USE_CLIENT_ID	1629	X'0000065D'
MQIACH_MQTT_KEEP_ALIVE	1630	X'0000065E'
MQIACH_IN_DOUBT_IN	1631	X'0000065F'
MQIACH_IN_DOUBT_OUT	1632	X'00000660'
MQIACH_MSGS_SENT<	1633	X'00000661'
MQIACH_MSGS_RECEIVED	1634	X'00000662'
MQIACH_MSGS_RCVD	1634	X'00000662'
MQIACH_PENDING_OUT	1635	X'00000663'
MQIACH_AVAILABLE_CIPHERSPECS	1636	X'00000664'
MQIACH_MATCH	1637	X'00000665'
MQIACH_USER_SOURCE	1638	X'00000666'
MQIACH_WARNING	1639	X'00000667'
MQIACH_DEF_RECONNECT	1640	X'00000668'
MQIACH_CHANNEL_SUMMARY_ATTRS	1642	X'0000066A'
MQIACH_PROTOCOL	1643	X'0000066B'
MQIACH_AMQPKEEPALIVE	1644	X'0000066C'
MQIACH_SECURITY_PROTOCOL	1645	X'0000066D'
 MQIACH_SPL_PROTECTION	1646	X'0000066E'
MQIACH_LAST_USED	1646	X'0000066E'

### **MQIAMO\_\* (Tipos de Parâmetros de Monitoramento de Número Inteiro do formato de Comando)**

Tabela 187. Valores de constantes

Nome	Valor decimal	Valor Hexadecimal
MQIAMO_FIRST	701	X'000002BD'
MQIAMO_AVG_BATCH_SIZE	702	X'000002BE'
MQIAMO_AVG_Q_TIME	703	X'000002BF'
MQIAMO_BACKOUTS	704	X'000002C0'
MQIAMO_BROWSES	705	X'000002C1'
MQIAMO_BROWSE_MAX_BYTES	706	X'000002C2'



Tabela 187. Valores de constantes (continuação)

Nome	Valor decimal	Valor Hexadecimal
MQIAMO_BROWSE_MIN_BYTES	707	X'000002C3'
MQIAMO_BROWSES_FAILED	708	X'000002C4'
MQIAMO_CLOSES	709	X'000002C5'
MQIAMO_COMMITS	710	X'000002C6'
MQIAMO_COMMITS_FAILED	711	X'000002C7'
MQIAMO_CONNS	712	X'000002C8'
MQIAMO_CONNS_MAX	713	X'000002C9'
MQIAMO_DISCS	714	X'000002CA'
MQIAMO_DISCS_IMPLICIT	715	X'000002CB'
MQIAMO_DISC_TYPE	716	X'000002CC'
MQIAMO_EXIT_TIME_AVG	717	X'000002CD'
MQIAMO_EXIT_TIME_MAX	718	X'000002CE'
MQIAMO_EXIT_TIME_MIN	719	X'000002CF'
MQIAMO_FULL_BATCHES	720	X'000002D0'
MQIAMO_GENERATED_MSGS	721	X'000002D1'
MQIAMO_GETS	722	X'000002D2'
MQIAMO_GET_MAX_BYTES	723	X'000002D3'
MQIAMO_GET_MIN_BYTES	724	X'000002D4'
MQIAMO_GETS_FAILED	725	X'000002D5'
MQIAMO_INCOMPLETE_BATCHES	726	X'000002D6'
MQIAMO_INQS	727	X'000002D7'
MQIAMO_MSGS	728	X'000002D8'
MQIAMO_NET_TIME_AVG	729	X'000002D9'
MQIAMO_NET_TIME_MAX	730	X'000002DA'
MQIAMO_NET_TIME_MIN	731	X'000002DB'
MQIAMO_OBJECT_COUNT	732	X'000002DC'
MQIAMO_OPENS	733	X'000002DD'
MQIAMO_PUT1S	734	X'000002DE'
MQIAMO_PUTS	735	X'000002DF'
MQIAMO_PUT_MAX_BYTES	736	X'000002E0'
MQIAMO_PUT_MIN_BYTES	737	X'000002E1'
MQIAMO_PUT_RETRIES	738	X'000002E2'
MQIAMO_Q_MAX_DEPTH	739	X'000002E3'
MQIAMO_Q_MIN_DEPTH	740	X'000002E4'
MQIAMO_Q_TIME_AVG	741	X'000002E5'
MQIAMO_Q_TIME_MAX	742	X'000002E6'
MQIAMO_Q_TIME_MIN	743	X'000002E7'
MQIAMO_SETS	744	X'000002E8'
MQIAMO_CONNS_FAILED	749	X'000002ED'

Tabela 187. Valores de constantes (continuação)

Nome	Valor decimal	Valor Hexadecimal
MQIAMO_OPENS_FAILED	751	X'000002EF'
MQIAMO_INQS_FAILED	752	X'000002F0'
MQIAMO_SETS_FAILED	753	X'000002F1'
MQIAMO_PUTS_FAILED	754	X'000002F2'
MQIAMO_PUT1S_FAILED	755	X'000002F3'
MQIAMO_CLOSES_FAILED	757	X'000002F5'
MQIAMO_MSGS_EXPIRED	758	X'000002F6'
MQIAMO_MSGS_NOT_QUEUED	759	X'000002F7'
MQIAMO_MSGS_PURGED	760	X'000002F8'
MQIAMO_SUBS_DUR	764	X'000002FC'
MQIAMO_SUBS_NDUR	765	X'000002FD'
MQIAMO_SUBS_FAILED	766	X'000002FE'
MQIAMO_SUBRQS	767	X'000002FF'
MQIAMO_SUBRQS_FAILED	768	X'00000300'
MQIAMO_CBS	769	X'00000301'
MQIAMO_CBS_FAILED	770	X'00000302'
MQIAMO_CTLs	771	X'00000303'
MQIAMO_CTLs_FAILED	772	X'00000304'
MQIAMO_STATS	773	X'00000305'
MQIAMO_STATS_FAILED	774	X'00000306'
MQIAMO_SUB_DUR_HIGHWATER	775	X'00000307'
MQIAMO_SUB_DUR_LOWWATER	776	X'00000308'
MQIAMO_SUB_NDUR_HIGHWATER	777	X'00000309'
MQIAMO_SUB_NDUR_LOWWATER	778	X'0000030A'
MQIAMO_TOPIC_PUTS	779	X'0000030B'
MQIAMO_TOPIC_PUTS_FAILED	780	X'0000030C'
MQIAMO_TOPIC_PUT1S	781	X'0000030D'
MQIAMO_TOPIC_PUT1S_FAILED	782	X'0000030E'
MQIAMO_PUBLISH_MSG_COUNT	784	X'00000310'
MQIAMO_UNSUBS_DUR	786	X'00000312'
MQIAMO_UNSUBS_NDUR	787	X'00000313'
MQIAMO_UNSUBS_FAILED	788	X'00000314'
MQIAMO_INTERVAL	789	X'00000315'
MQIAMO_MSGS_SENT	790	X'00000316'
MQIAMO_BYTES_SENT	791	X'00000317'
MQIAMO_REPAIR_BYTES	792	X'00000318'
MQIAMO_FEEDBACK_MODE	793	X'00000319'
MQIAMO_RELIABILITY_TYPE	794	X'0000031A'
MQIAMO_LATE_JOIN_MARK	795	X'0000031B'

Tabela 187. Valores de constantes (continuação)

Nome	Valor decimal	Valor Hexadecimal
MQIAMO_NACKS_RCVD	796	X'0000031C'
MQIAMO_REPAIR_PKTS	797	X'0000031D'
MQIAMO_HISTORY_PKTS	798	X'0000031E'
MQIAMO_PENDING_PKTS	799	X'0000031F'
MQIAMO_PKT_RATE	800	X'00000320'
MQIAMO_MCAST_XMIT_RATE	801	X'00000321'
MQIAMO_MCAST_BATCH_TIME	802	X'00000322'
MQIAMO_MCAST_HEARTBEAT	803	X'00000323'
MQIAMO_DEST_DATA_PORT	804	X'00000324'
MQIAMO_DEST_REPAIR_PORT	805	X'00000325'
MQIAMO_ACKS_RCVD	806	X'00000326'
MQIAMO_ACTIVE_ACKERS	807	X'00000327'
MQIAMO_PKTS_SENT	808	X'00000328'
MQIAMO_TOTAL_REPAIR_PKTS	809	X'00000329'
MQIAMO_TOTAL_PKTS_SENT	810	X'0000032A'
MQIAMO_TOTAL_MSGS_SENT	811	X'0000032B'
MQIAMO_TOTAL_BYTES_SENT	812	X'0000032C'
MQIAMO_NUM_STREAMS	813	X'0000032D'
MQIAMO_ACK_FEEDBACK	814	X'0000032E'
MQIAMO_NACK_FEEDBACK	815	X'0000032F'
MQIAMO_PKTS_LOST	816	X'00000330'
MQIAMO_MSGS_RCVD	817	X'00000331'
MQIAMO_MSG_BYTES_RCVD	818	X'00000332'
MQIAMO_MSGS_ENTREGUES	819	X'00000333'
MQIAMO_PKTS_PROCESSED	820	X'00000334'
MQIAMO_PKTS_DLVD	821	X'00000335'
MQIAMO_PKTS_DROPPED	822	X'00000336'
MQIAMO_PKTS_DUPLICATED	823	X'00000337'
MQIAMO_NACKS_CREATED	824	X'00000338'
MQIAMO_NACK_PKTS_SENT	825	X'00000339'
MQIAMO_REPAIR_PKTS_RQSTD	826	X'0000033A'
MQIAMO_REPAIR_PKTS_RCVD	827	X'0000033B'
MQIAMO_PKTS_REPAIRED	828	X'0000033C'
MQIAMO_TOTAL_MSGS_RCVD	829	X'0000033D'
MQIAMO_TOTAL_MSGS_BYTES_RCVD	830	X'0000033E'
MQIAMO_TOTAL_REPAIR_PKTS_RCVD	831	X'0000033F'
MQIAMO_TOTAL_REPAIR_PKTS_RQSTD	832	X'00000340'
MQIAMO_TOTAL_MSGS_PROCESSED	833	X'00000341'
MQIAMO_TOTAL_MSGS_SELECTED	834	X'00000342'

Tabela 187. Valores de constantes (continuação)

Nome	Valor decimal	Valor Hexadecimal
MQIAMO_TOTAL_MSGS_EXPIRADO	835	X'00000343'
MQIAMO_TOTAL_MSGS_ENTREGUES	836	X'00000344'
MQIAMO_TOTAL_MSGS_RETORNADO	837	X'00000345'
MQIAMO_LAST_USED	837	X'00000345'

### **MQIAMO64\_\* (Tipos de Parâmetro de Monitoramento de Número Inteiro de 64 bits)**

Tabela 188. Valores de constantes

Nome	Valor decimal	Valor Hexadecimal
MQIAMO64_AVG_Q_TIME	703	X'000002BF'
MQIAMO64_Q_TIME_AVG	741	X'000002E5'
MQIAMO64_Q_TIME_MAX	742	X'000002E6'
MQIAMO64_Q_TIME_MIN	743	X'000002E7'
MQIAMO64_BROWSE_BYTES	745	X'000002E9'
MQIAMO64_BYTES	746	X'000002EA'
MQIAMO64_GET_BYTES	747	X'000002EB'
MQIAMO64_PUT_BYTES	748	X'000002EC'
MQIAMO64_TOPIC_PUT_BYTES	783	X'0000030F'
MQIAMO64_PUBLISH_MSG_BYTES	785	X'00000311'

### **MQIASY\_\* (Seletores de sistema de número inteiro)**

Tabela 189. Valores de constantes

Nome	Valor decimal	Valor Hexadecimal
MQIASY_FIRST	-1	X'FFFFFFFF'
MQIASY_CODED_CHAR_SET_ID	-1	X'FFFFFFFF'
MQIASY_TYPE	-2	X'FFFFFFFE'
MQIASY_COMMAND	-3	X'FFFFFFFD'
MQIASY_MSG_SEQ_NUMBER	-4	X'FFFFFFFC'
MQIASY_CONTROL	-5	X'FFFFFFFB'
MQIASY_COMP_CODE	-6	X'FFFFFFFA'
MQIASY_REASON	-7	X'FFFFFFF9'
MQIASY_BAG_OPTIONS	-8	X'FFFFFFF8'
MQIASY_VERSION	-9	X'FFFFFFF7'
MQIASY_LAST_USED	-9	X'FFFFFFF7'
MQIASY_LAST	-2000	X'FFFFFF830'

### **MQIAUT\_\* (Autenticador do cabeçalho de informações IMS)**

Tabela 190. Nomes e valores de constantes

Nome	Value
MQIAUT_NONE	"rrrrrrrr"

Tabela 190. Nomes e valores de constantes (continuação)	
Nome	Value
MQIAUT_NONE_ARRAY	'¬', '¬', '¬', '¬', '¬', '¬', '¬', '¬', '¬', '¬'

**Nota:** O símbolo ¬ representa um único caractere em branco.

### **MQIAV\_\* (Valores de atributo de número inteiro)**

Tabela 191. Valores de constantes		
Nome	Valor decimal	Valor Hexadecimal
MQIAV_NOT_APLICÁVEL	-1	X'FFFFFFFF'
MQIAV_UNDEFINED	-2	X'FFFFFFFE'

### **MQICM\_\* (IMS modos de confirmação do cabeçalho de informações)**

Tabela 192. Nomes e valores de constantes	
Nome	Value
MQICM_COMMIT_THEN_SEND	'0'
MQICM_SEND_THEN_COMMIT	'1'

### **MQIDO\_\* (Opções Indeterminadas de Formato de Comando).**

Tabela 193. Valores de constantes		
Nome	Valor decimal	Valor Hexadecimal
MQIDO_COMMIT	1	X'00000001'
MQIDO_BACKOUT	2	X'00000002'

### **MQIEP\_\* (Pontos de Entrada da Interface).**

#### **Estrutura dos parâmetros de segurança da conexão**

Tabela 194. Estruturas de constantes	
Nome	Estrutura
MQIEP_STRUC_ID	"IEP¬"
MQIEP_STRUC_ID_ARRAY	'I', 'E', 'P', '¬'

**Nota:** O símbolo ¬ representa um único caractere em branco.

Tabela 195. Valores de constantes		
Nome	Valor decimal	Valor Hexadecimal
MQIEP_VERSION_1	1	X'00000001'
MQDXP_CURRENT_VERSION	1	X'00000001'

### **MQIGQ\_\* (Enfileiramento Intra-Group)**

Tabela 196. Valores de constantes		
Nome	Valor decimal	Valor Hexadecimal
MQIGQ_DISABLED	0	X'00000000'
MQIGQ_ENABLED	1	X'00000001'

## MQIGQPA\_\* (Autoridade de Colocação no Enfileiramento do Grupo)

Tabela 197. Valores de constantes		
Nome	Valor decimal	Valor Hexadecimal
MQIGQPA_DEFAULT	1	X'00000001'
MQIGQPA_CONTEXT	2	X'00000002'
MQIGQPA_ONLY_IGQ	3	X'00000003'
MQIGQPA_ALTERNATE_OR_IGQ	4	X'00000004'

## MQIIH\_\* (estrutura do cabeçalho de informações IMS e Sinalizadores)

### Estrutura do cabeçalho de informações IMS

Tabela 198. Estruturas de constantes	
Nome	Estrutura
MQIIH_STRUC_ID	"IIH~"
MQIIH_STRUC_ID_ARRAY	'I','I','H','~'

**Nota:** O símbolo ~ representa um único caractere em branco.

Tabela 199. Valores de constantes		
Nome	Valor decimal	Valor Hexadecimal
MQIIH_VERSION_1	1	X'00000001'
MQIIH_CURRENT_VERSION	1	X'00000001'
MQIIH_LENGTH_1	84	X'00000054'

### Sinalizadores do cabeçalho de informações IMS

Tabela 200. Valores de constantes		
Nome	Valor decimal	Valor Hexadecimal
MQIIH_NONE	0	X'00000000'
MQIIH_PASS_EXPIRATION	1	X'00000001'
MQIIH_UNLIMITED_EXPIRATION	0	X'00000000'
MQIIH_REPLY_FORMAT_NONE	8	X'00000008'
MQIIH_IGNORE_PURG	16	X'00000010'
MQIIH_CM0_REQUEST_RESPONSE	32	X'00000020'

## MQIMPO\_\* (opções e estrutura da propriedade de mensagem de Consulta).

### Consultar estrutura de opções de propriedades de mensagens

Tabela 201. Estruturas de constantes	
Nome	Estrutura
MQIMPO_STRUC_ID	"IMPO"
MQIMPO_STRUC_ID_ARRAY	'I','M','P','O'

**Nota:** O símbolo ~ representa um único caractere em branco.

<i>Tabela 202. Valores de constantes</i>		
Nome	Valor decimal	Valor Hexadecimal
MQIMPO_VERSION_1	1	X'00000001'
MQIMPO_CURRENT_VERSION	1	X'00000001'

### Consultar Opções de Propriedade de Mensagem

<i>Tabela 203. Valores de constantes</i>		
Nome	Valor decimal	Valor Hexadecimal
MQIMPO_CONVERT_TYPE	2	X'00000002'
MQIMPO_QUERY_LENGTH	4	X'00000004'
MQIMPO_INQ_FIRST	0	X'00000000'
MQIMPO_INQ_NEXT	8	X'00000008'
MQIMPO_INQ_PROP_UNDER_CURSOR	16	X'00000010'
MQIMPO_CONVERT_VALUE	32	X'00000020'
MQIMPO_NONE	0	X'00000000'

### MQINBD\_\* (Disposições de entrada do formato de comando)

<i>Tabela 204. Valores de constantes</i>		
Nome	Valor decimal	Valor Hexadecimal
MQINBD_Q_MGR	0	X'00000000'
MQINBD_GROUP	3	X'00000003'

### MQIND\_\* (Valores de Índice Especial)

<i>Tabela 205. Valores de constantes</i>		
Nome	Valor decimal	Valor Hexadecimal
MQIND_NONE	-1	X'FFFFFFFF'
MQIND_ALL	-2	X'FFFFFFFE'

### MQIPADDR\_\* (versões de endereço IP)

<i>Tabela 206. Valores de constantes</i>		
Nome	Valor decimal	Valor Hexadecimal
MQIPADDR_IPV4	0	X'00000000'
MQIPADDR_IPV6	1	X'00000001'

### MQISS\_\* (Escopos de Segurança do cabeçalho de informação IMS)

<i>Tabela 207. Nomes e valores de constantes</i>	
Nome	Value
MQISS_CHECK	'C'
MQISS_FULL	'F'

## MQIT\_\* (Tipos de Índice).

Tabela 208. Valores de constantes		
Nome	Valor decimal	Valor Hexadecimal
MQIT_NONE	0	X'00000000'
MQIT_MSG_ID	1	X'00000001'
MQIT_CORREL_ID	2	X'00000002'
MQIT_MSG_TOKEN	4	X'00000004'
MQIT_GROUP_ID	5	X'00000005'

## MQITEM\_\* (Tipo de Item para mqInquireItemInfo)

Tabela 209. Valores de constantes		
Nome	Valor decimal	Valor Hexadecimal
MQITEM_INTEGER	1	X'00000001'
MQITEM_STRING	2	X'00000002'
MQITEM_BAG	3	X'00000003'
MQITEM_BYTE_STRING	4	X'00000004'
MQITEM_INTEGER_FILTER	5	X'00000005'
MQITEM_STRING_FILTER	6	X'00000006'
MQITEM_INTEGER64	7	X'00000007'
MQITEM_BYTE_STRING_FILTER	8	X'00000008'

## MQITII\_\* (Identificador da instância de transação do cabeçalho de informações IMS)

Tabela 210. Nomes e valores de constantes	
Nome	Value
MQITII_NONE	X'00...00' (16 nulos)
MQITII_NONE_ARRAY	'\0', '\0', ... (16 nulos)

## MQITS\_\* (IMS estados da transação do cabeçalho de informações)

Tabela 211. Nomes e valores de constantes	
Nome	Value
MQITS_IN_CONVERSATION	'C'
MQITS_NOT_IN_CONVERSATION	'-'
MQITS_ARCHITECTED	'A'

**Nota:** O símbolo - representa um único caractere em branco.

## MQKAI\_\* (IntervaloKeepAlive)

Tabela 212. Valores de constantes		
Nome	Valor decimal	Valor Hexadecimal
MQKAI_AUTO	-1	X'FFFFFFFF'



## MQMASTER\_\* (Administração principal)

Tabela 213. Valores de constantes		
Nome	Valor decimal	Valor Hexadecimal
MQMASTER_NO	0	X'00000000'
MQMASTER_YES	1	X'00000001'

## MQMCAS\_\* (Status do Agente do Canal de Mensagem do formato de comando)...

Tabela 214. Valores de constantes		
Nome	Valor decimal	Valor Hexadecimal
MQMCAS_STOPPED	0	X'00000000'
MQMCAS_RUNNING	3	X'00000003'

## MQMCAT\_\* (Tipos MCA)

Tabela 215. Valores de constantes		
Nome	Valor decimal	Valor Hexadecimal
MQMCAT_PROCESS	1	X'00000001'
MQMCAT_THREAD	2	X'00000002'

## MQMCD\_\* (Informações da Tag Opções de Publicação / Assinatura)

### Tags do Descritor de Conteúdo da Mensagem da Tag de Opções de Publicação / Assinatura (mcd)

Tabela 216. Valores de constantes		
Nome	Valor decimal	Valor Hexadecimal
MQMCD_FOLDER_VERSION	1	X'00000001'

## Nomes de Tag de Opções de Publicação / Assinatura

Tabela 217. Nomes e valores de constantes	
Nome	Value
MQMCD_MSG_DOMAIN	"Msd"
MQMCD_MSG_SET	"Set"
MQMCD_MSG_TYPE	"Type"
MQMCD_MSG_FORMAT	"Fmt"

## Nomes de tags XML de tag de opções de publicação / assinatura

Tabela 218. Nomes e valores de constantes	
Nome	Value
MQMCD_MSG_DOMAIN_B	"<Msd>"
MQMCD_MSG_DOMAIN_E	"</Msd>"
MQMCD_MSG_SET_B	"<Set>"
MQMCD_MSG_SET_E	"</Set>"

Tabela 218. Nomes e valores de constantes (continuação)	
Nome	Value
MQMCD_MSG_TYPE_B	"<Type>"
MQMCD_MSG_TYPE_E	"</Type>"
MQMCD_MSG_FORMAT_B	"<Fmt>"
MQMCD_MSG_FORMAT_E	"</Fmt>"

### Valores da Tag de Opções de Publicação / Assinatura

Tabela 219. Nomes e valores de constantes	
Nome	Value
MQMCD_DOMAIN_NONE	"none"
MQMCD_DOMAIN_NEON	"neon"
MQMCD_DOMAIN_MRM	"mrm"
MQMCD_DOMAIN_JMS_NONE	"jms_none"
MQMCD_DOMAIN_JMS_TEXT	"jms_text"
MQMCD_DOMAIN_JMS_OBJECT	"jms_object"
MQMCD_DOMAIN_JMS_MAP	"jms_map"
MQMCD_DOMAIN_JMS_STREAM	"jms_stream"
MQMCD_DOMAIN_JMS_BYTES	"jms_bytes"

### MQMD\_\* (Estrutura do descritor de mensagens)

Tabela 220. Estruturas de constantes	
Nome	Estrutura
MQMD_STRUC_ID	"MD↵"
MQMD_STRUC_ID_ARRAY	'M', 'D', '↵', '↵'

**Nota:** O símbolo ↵ representa um único caractere em branco.

Tabela 221. Valores de constantes		
Nome	Valor decimal	Valor Hexadecimal
MQMD_VERSION_1	1	X'00000001'
MQMD_VERSION_2	2	X'00000002'
MQMD_CURRENT_VERSION	2	X'00000002'

### MQMDE\_\* (Estrutura de extensão do descritor de mensagens)

Tabela 222. Estruturas de constantes	
Nome	Estrutura
MQMDE_STRUC_ID	"MDE↵"
MQMDE_STRUC_ID_ARRAY	'M', 'D', 'E', '↵'

**Nota:** O símbolo ↵ representa um único caractere em branco.

Tabela 223. Valores de constantes		
Nome	Valor decimal	Valor Hexadecimal
MQMDE_VERSION_2	2	X'00000002'
MQMDE_CURRENT_VERSION	2	X'00000002'
MQMDE_LENGTH_2	72	X'00000048'

### MQMDEF\_\* (Sinalizadores de extensão do descritor de mensagens)

Tabela 224. Valores de constantes		
Nome	Valor decimal	Valor Hexadecimal
MQMDEF_NONE	0	X'00000000'

### MQMDS\_\* (Sequência de Entrega de Mensagens)..

Tabela 225. Valores de constantes		
Nome	Valor decimal	Valor Hexadecimal
MQMDS_PRIORITY	0	X'00000000'
MQMDS_FIFO	1	X'00000001'

### MQMF\_\* (Sinalizações de Mensagem)

Tabela 226. Valores de constantes		
Nome	Valor decimal	Valor Hexadecimal
MQMF_SEGMENTATION_INIBIDA	0	X'00000000'
MQMF_SEGMENTATION_ALLOWED	1	X'00000001'
MQMF_MSG_IN_GROUP	8	X'00000008'
MQMF_LAST_MSG_IN_GROUP	16	X'00000010'
MQMF_SEGMENT	2	X'00000002'
MQMF_LAST_SEGMENT	4	X'00000004'
MQMF_NONE	0	X'00000000'

### MQMHBO\_\* (Identificador de mensagens para opções de buffer e estrutura).

#### Identificador de mensagem para estrutura de opções de buffer

Tabela 227. Estruturas de constantes	
Nome	Estrutura
MQMHBO_STRUC_ID	"MHBO"
MQMHBO_STRUC_ID_ARRAY	'M', 'H', 'B', 'O'

**Nota:** O símbolo ~ representa um único caractere em branco.

Tabela 228. Valores de constantes		
Nome	Valor decimal	Valor Hexadecimal
MQMHBO_VERSION_1	1	X'00000001'
MQMHBO_CURRENT_VERSION	1	X'00000001'

## Manipulação de mensagem para opções de buffer

*Tabela 229. Valores de constantes*

Nome	Valor decimal	Valor Hexadecimal
MQMHBO_PROPERTIES_IN_MQRFH2	1	X'00000001'
MQMHBO_DELETE_PROPERTIES	2	X'00000002'
MQMHBO_NONE	0	X'00000000'

## MQMI\_\* (Identificador de mensagem)..

*Tabela 230. Nomes e valores de constantes*

Nome	Value
MQMI_NONE	X'00...00' (24 nulos)
MQMI_NONE_ARRAY	'\0', '\0', ... (24 nulos)

## MQMMBI\_\* (Marca da Mensagem-Intervalo de Procura)

*Tabela 231. Valores de constantes*

Nome	Valor decimal	Valor Hexadecimal
MQMMBI_UNLIMITED	-1	X'FFFFFFFF'

## MQMO\_\* (Opções de Correspondência)

*Tabela 232. Valores de constantes*

Nome	Valor decimal	Valor Hexadecimal
MQMO_MATCH_MSG_ID	1	X'00000001'
MQMO_MATCH_CORREL_ID	2	X'00000002'
MQMO_MATCH_GROUP_ID	4	X'00000004'
MQMO_MATCH_MSG_SEQ_NUMBER	8	X'00000008'
MQMO_MATCH_OFFSET	16	X'00000010'
MQMO_MATCH_MSG_TOKEN	32	X'00000020'
MQMO_NONE	0	X'00000000'

## MQMODE\_\* (Opções do Modo de Formato de Comandos)..

*Tabela 233. Valores de constantes*

Nome	Valor decimal	Valor Hexadecimal
MQMODE_FORCE	0	X'00000000'
MQMODE QUIESCE	1	X'00000001'
MQMODE_TERMINATE	2	X'00000002'

## MQMON\_\* (Valores de Monitoramento)

*Tabela 234. Valores de constantes*

Nome	Valor decimal	Valor Hexadecimal
MQMON_NOT_AVAILABLE	-1	X'FFFFFFFF'
MQMON_NONE	-1	X'FFFFFFFF'

Tabela 234. Valores de constantes (continuação)

Nome	Valor decimal	Valor Hexadecimal
MQMON_Q_MGR	-3	X'FFFFFFFD'
MQMON_OFF	0	X'00000000'
MQMON_ON	1	X'00000001'
MQMON_DISABLED	0	X'00000000'
MQMON_ENABLED	1	X'00000001'
MQMON_LOW	17	X'00000011'
MQMON_MEDIUM	33	X'00000021'
MQMON_HIGH	65	X'00000041'

### MQMT\_\* (Tipos de Mensagens).

Tabela 235. Valores de constantes

Nome	Valor decimal	Valor Hexadecimal
MQMT_SYSTEM_FIRST	1	X'00000001'
MQMT_REQUEST	1	X'00000001'
MQMT_REPLY	2	X'00000002'
MQMT_DATAGRAM	8	X'00000008'
MQMT_REPORT	4	X'00000004'
MQMT_MQE_FIELDS_FROM_MQE	112	X'00000070'
MQMT_MQE_FIELDS	113	X'00000071'
MQMT_SYSTEM_LAST	65535	X'0000FFFF'
MQMT_APPL_FIRST	65536	X'00010000'
MQMT_APPL_LAST	99999999	X'3B9AC9FF'

### MQMTOK\_\* (Token de mensagem)

Tabela 236. Nomes e valores de constantes

Nome	Value
MQMTOK_NONE	X'00...00' (16 nulos)
MQMTOK_NONE_ARRAY	'\0', '\0', ... (16 nulos)

### MQNC\_\* (contagem de nomes)

Tabela 237. Valores de constantes

Nome	Valor decimal	Valor Hexadecimal
MQNC_MAX_NAMELIST_NAME_COUNT	256	X'00000100'

### MQNPM\_\* (Classe de mensagem não persistente)

Tabela 238. Valores de constantes

Nome	Valor decimal	Valor Hexadecimal
MQNPM_CLASS_NORMAL	0	X'00000000'
MQNPM_CLASS_HIGH	10	X'0000000A'

## MQNPMS\_\* (NonPersistent-Message Speeds)

Tabela 239. Valores de constantes		
Nome	Valor decimal	Valor Hexadecimal
MQNPMS_NORMAL	1	X'00000001'
MQNPMS_FAST	2	X'00000002'

## MQNT\_\* (Tipos de lista de nomes).

Tabela 240. Valores de constantes		
Nome	Valor decimal	Valor Hexadecimal
MQNT_NONE	0	X'00000000'
MQNT_Q	1	X'00000001'
MQNT_CLUSTER	2	X'00000002'
MQNT_AUTH_INFO	4	X'00000004'
MQNT_ALL	1001	X'000003E9'

## MQNVS\_\* (Nomes para Sequência de Nome / Valor)

Tabela 241. Nomes e valores de constantes	
Nome	Value
MQNVS_APPL_TYPE	"OPT_APP_GRP↵"
MQNVS_MSG_TYPE	"OPT_MSG_TYPE↵"

**Nota:** O símbolo ↵ representa um único caractere em branco.

## MQOA\_\* (Limites para Seletores para Atributos de Objeto)..

Tabela 242. Valores de constantes		
Nome	Valor decimal	Valor Hexadecimal
MQOA_FIRST	1	X'00000001'
MQOA_LAST	9000	X'00002328'

## MQOD\_\* (Estrutura do descritor de objeto)

Tabela 243. Estruturas de constantes	
Nome	Estrutura
MQOD_STRUC_ID	"OD↵"
MQOD_STRUC_ID_ARRAY	'0', 'D', '↵', '↵'

**Nota:** O símbolo ↵ representa um único caractere em branco.

Tabela 244. Valores de constantes		
Nome	Valor decimal	Valor Hexadecimal
MQOD_VERSION_1	1	X'00000001'
MQOD_VERSION_2	2	X'00000002'
MQOD_VERSION_3	3	X'00000003'
MQOD_VERSION_4	4	X'00000004'

Tabela 244. Valores de constantes (continuação)		
Nome	Valor decimal	Valor Hexadecimal
MQOD_CURRENT_VERSION	4	X'00000004'
MQOD_CURRENT_LENGTH	(value differs by platform or version)	(value differs by platform or version)

### MQOII\_\* (Identificador de instância de objeto).

Tabela 245. Nomes e valores de constantes	
Nome	Value
MQOII_NONE	X'00...00' (24 nulos)
MQOII_NONE_ARRAY	'\0', '\0', ... (24 nulos)

### MQOL\_\* (Comprimento original)

Tabela 246. Valores de constantes		
Nome	Valor decimal	Valor Hexadecimal
MQOL_UNDEFINED	-1	X'FFFFFFFF'

### MQOM\_\* (Opções de Mensagens Db2 Obsoletas no Grupo de Consulta)

Tabela 247. Valores de constantes		
Nome	Valor decimal	Valor Hexadecimal
MQOM_NO	0	X'00000000'
MQOM_YES	1	X'00000001'

### MQOO\_\* (Opções de abertura)

Tabela 248. Valores de constantes		
Nome	Valor decimal	Valor Hexadecimal
MQOO_BIND_AS_Q_DEF	0	X'00000000'
MQOO_READ_AHEAD_AS_Q_DEF	0	X'00000000'
MQOO_INPUT_AS_Q_DEF	1	X'00000001'
MQOO_INPUT_SHARED	2	X'00000002'
MQOO_INPUT_EXCLUSIVE	4	X'00000004'
MQOO_BROWSE	8	X'00000008'
MQOO_OUTPUT	16	X'00000010'
MQOO_INQUIRE	32	X'00000020'
MQOO_SET	64	X'00000040'
MQOO_SAVE_ALL_CONTEXT	128	X'00000080'
MQOO_PASS_IDENTITY_CONTEXT	256	X'00000100'
MQOO_PASS_ALL_CONTEXT	512	X'00000200'
MQOO_SET_IDENTITY_CONTEXT	1024	X'00000400'
MQOO_SET_ALL_CONTEXT	2048	X'00000800'
MQOO_ALTERNATE_USER_AUTHORITY	4096	X'00001000'
MQOO_FAIL_IF QUIESCING	8192	X'00002000'

Tabela 248. Valores de constantes (continuação)		
Nome	Valor decimal	Valor Hexadecimal
MQOO_BIND_ON_OPEN	16384	X'00004000'
MQOO_BIND_NOT_FIXED	32768	X'00008000'
MQOO_CO_OP	131072	X'00020000'
MQOO_RESOLVE_LOCAL_TOPIC	262144	X'00040000'
MQOO_NO_READ_AHEAD	524288	X'00080000'
MQOO_READ_AHEAD	1048576	X'00100000'
MQOO_BIND_ON_GROUP	4194304	X'00400000'

### ***MQOO\_\** (Seguindo usado em C++ apenas)**

Tabela 249. Valores de constantes		
Nome	Valor decimal	Valor Hexadecimal
MQOO_RESOLVE_NAMES	65536	X'00010000'
MQOO_RESOLVE_LOCAL_Q	262144	X'00040000'

### ***MQOP\_\** (códigos de operação para MQCTL e MQCB)**

#### **Códigos de operação para MQCTL**

Tabela 250. Valores de constantes		
Nome	Valor decimal	Valor Hexadecimal
MQOP_START	1	X'00000001'
MQOP_START_WAIT	2	X'00000002'
MQOP_STOP	4	X'00000004'

#### **Códigos de Operação para MQCB**

Tabela 251. Valores de constantes		
Nome	Valor decimal	Valor Hexadecimal
MQOP_REGISTER	256	X'00000100'
MQOP_REMOVER registro	512	X'00000200'

#### **Códigos de operação para MQCTL/MQCB**

Tabela 252. Valores de constantes		
Nome	Valor decimal	Valor Hexadecimal
MQOP_SUSPEND	65536	X'00010000'
MQOP_RESUME	131072	X'00020000'

### ***MQOPEN\_\** (Valores relacionados à estrutura de MQOPEN\_PRIV)**

Tabela 253. Valores de constantes		
Nome	Valor decimal	Valor Hexadecimal
MQOPEN_PRIV_VERSION_1	1	X'00000001'
MQOPEN_PRIV_CURRENT_VERSION	1	X'00000001'



## MQOPER\_\* (Operações de atividade)

*Tabela 254. Valores de constantes*

Nome	Valor decimal	Valor Hexadecimal
MQOPER_SYSTEM_FIRST	0	X'00000000'
MQOPER_UNKNOWN	0	X'00000000'
MQOPER_BROWSE	1	X'00000001'
MQOPER_DISCARD	2	X'00000002'
MQOPER_GET	3	X'00000003'
MQOPER_PUT	4	X'00000004'
MQOPER_PUT_REPLY	5	X'00000005'
MQOPER_PUT_REPORT	6	X'00000006'
MQOPER_RECEIVE	7	X'00000007'
MQOPER_SEND	8	X'00000008'
MQOPER_TRANSFORM	9	X'00000009'
MQOPER_PUBLISH	10	X'0000000A'
MQOPER_EXCLUDED_PUBLISH	11	X'0000000B'
MQOPER_DISCARDED_PUBLISH	12	X'0000000C'
MQOPER_SYSTEM_LAST	65535	X'0000FFFF'
MQOPER_APPL_FIRST	65536	X'00010000'
MQOPER_APPL_LAST	999999999	X'3B9AC9FF'

## MQOT\_\* (Tipos de Objeto e Tipos de Objeto Estendido)

### Tipos de objetos

*Tabela 255. Valores de constantes*

Nome	Valor decimal	Valor Hexadecimal
MQOT_NONE	0	X'00000000'
MQOT_Q	1	X'00000001'
MQOT_NAMELIST	2	X'00000002'
MQOT_PROCESS	3	X'00000003'
MQOT_STORAGE_CLASS	4	X'00000004'
MQOT_Q_MGR	5	X'00000005'
MQOT_CHANNEL	6	X'00000006'
MQOT_AUTH_INFO	7	X'00000007'
MQOT_TOPIC	8	X'00000008'
MQOT_CF_STRUC	10	X'0000000A'
MQOT_LISTENER	11	X'0000000B'
MQOT_SERVICE	12	X'0000000C'
MQOT_RESERVED_1	999	X'000003E7'

## Tipos de Objeto Estendido

<i>Tabela 256. Valores de constantes</i>		
Nome	Valor decimal	Valor Hexadecimal
MQOT_ALL	1001	X'000003E9'
MQOT_ALIAS_Q	1002	X'000003EA'
MQOT_MODEL_Q	1003	X'000003EB'
MQOT_LOCAL_Q	1004	X'000003EC'
MQOT_REMOTE_Q	1005	X'000003ED'
MQOT_SENDER_CHANNEL	1007	X'000003EF'
MQOT_SERVER_CHANNEL	1008	X'000003F0'
MQOT_REQUESTER_CHANNEL	1009	X'000003F1'
MQOT_RECEIVER_CHANNEL	1010	X'000003F2'
MQOT_CURRENT_CHANNEL	1011	X'000003F3'
MQOT_SAVED_CHANNEL	1012	X'000003F4'
MQOT_SVRCONN_CHANEXOL	1013	X'000003F5'
MQOT_CLNTCONN_CHANEXOL	1014	X'000003F6'
MQOT_SHORT_CHANNEL	1015	X'000003F7'
MQOT_CHLAUTH	1016	X'000003F8'
MQOT_REMOTE_Q_MGR_NAME	1017	X'000003F9'
MQOT_PROT_POLICY	1019	X'000003FB'
MQOT_TT_CHANNEL	1020	X'000003FC'
MQOT_AMQP_CHANEXOL	1021	X'000003FD'
MQOT_AUTH_REC	1022	X'000003FE'

## MQPA\_\* (autoridade de colocação)

<i>Tabela 257. Valores de constantes</i>		
Nome	Valor decimal	Valor Hexadecimal
MQPA_DEFAULT	1	X'00000001'
MQPA_CONTEXT	2	X'00000002'
MQPA_ONLY_MCA	3	X'00000003'
MQPA_ALTERNATE_OR_MCA	4	X'00000004'

## MQPD\_\* (Descritor de propriedades, suporte e contexto)

### estrutura do descritor de propriedade

<i>Tabela 258. Estruturas de constantes</i>	
Nome	Estrutura
MQPD_STRUC_ID	"PD↵"
MQPD_STRUC_ID_ARRAY	'P','D','↵','↵'

**Nota:** O símbolo ↵ representa um único caractere em branco.

<i>Tabela 259. Valores de constantes</i>		
Nome	Valor decimal	Valor Hexadecimal
MQPD_VERSION_1	1	X'00000001'
MQPD_CURRENT_VERSION	1	X'00000001'

### Opções do Descritor de Propriedade.

<i>Tabela 260. Valores de constantes</i>		
Nome	Valor decimal	Valor Hexadecimal
MQPD_NONE	0	X'00000000'

### Opções de suporte de propriedade

<i>Tabela 261. Valores de constantes</i>		
Nome	Valor decimal	Valor Hexadecimal
MQPD_SUPPORT_OPTIONAL	1	X'00000001'
MQPD_SUPPORT_REQUIRED	1048576	X'00100000'
MQPD_SUPPORT_REQUIRED_IF_LOCAL	1024	X'00000400'
MQPD_REJECT_UNSUP_MASK	-1048576	X'FFF00000'
MQPD_ACCEPT_UNSUP_IF_XMIT_MASK	1047552	X'000FFC00'
MQPD_ACCEPT_UNSUP_MASK	1023	X'000003FF'

### Contexto da Propriedade.

<i>Tabela 262. Valores de constantes</i>		
Nome	Valor decimal	Valor Hexadecimal
MQPD_NO_CONTEXT	0	X'00000000'
MQPD_USER_CONTEXT	1	X'00000001'

### MQPER\_\* (Valores de Persistência)

<i>Tabela 263. Valores de constantes</i>		
Nome	Valor decimal	Valor Hexadecimal
MQPER_PERSISTENCE_AS_PARENT	-1	X'FFFFFFFF'
MQPER_NOT_PERSISTENT	0	X'00000000'
MQPER_PERSISTENT	1	X'00000001'
MQPER_PERSISTENCE_AS_Q_DEF	2	X'00000002'
MQPER_PERSISTENCE_AS_TOPIC_DEF	2	X'00000002'

### MQPL\_\* (Plataformas)

<i>Tabela 264. Valores de constantes</i>		
Nome	Valor decimal	Valor Hexadecimal
MQPL_MVS	1	X'00000001'
MQPL_OS390	1	X'00000001'
MQPL_ZOS	1	X'00000001'

*Tabela 264. Valores de constantes (continuação)*

Nome	Valor decimal	Valor Hexadecimal
MQPL_OS2	2	X'00000002'
MQPL_AIX	3	X'00000003'
MQPL_UNIX	3	X'00000003'
MQPL_OS400	4	X'00000004'
MQPL_WINDOWS	5	X'00000005'
MQPL_WINDOWS_NT	11	X'0000000B'
MQPL_VMS	12	X'0000000C'
MQPL_NSK	13	X'0000000D'
MQPL_OPEN_TP1	15	X'0000000F'
MQPL_VM	18	X'00000012'
MQPL_TPF	23	X'00000017'
MQPL_VSE	27	X'0000001B'
MQPL_APPLIANCE	28	X'0000001C'
MQPL_NATIVE	1	X'00000001'

## MQPMO\_\* (Colocar opções de mensagem e estrutura para máscara de publicação)

### Estrutura de opções de mensagem put

*Tabela 265. Estruturas de constantes*

Nome	Estrutura
MQPMO_STRUC_ID	"PMO~"
MQPMO_STRUC_ID_ARRAY	'P', 'M', 'O', '~'

**Nota:** O símbolo ~ representa um único caractere em branco.

*Tabela 266. Valores de constantes*

Nome	Valor decimal	Valor Hexadecimal
MQPMO_VERSION_1	1	X'00000001'
MQPMO_VERSION_2	2	X'00000002'
MQPMO_VERSION_3	3	X'00000003'
MQPMO_CURRENT_VERSION	3	X'00000003'
MQPMO_CURRENT_LENGTH	(value differs by platform or version)	(value differs by platform or version)

### Colocar opções de mensagem

*Tabela 267. Valores de constantes*

Nome	Valor decimal	Valor Hexadecimal
MQPMO_SYNCPOINT	2	X'00000002'
MQPMO_NO_SYNCPOINT	4	X'00000004'
MQPMO_DEFAULT_CONTEXT	32	X'00000020'

*Tabela 267. Valores de constantes (continuação)*

<b>Nome</b>	<b>Valor decimal</b>	<b>Valor Hexadecimal</b>
MQPMO_NEW_MSG_ID	64	X'00000040'
MQPMO_NEW_CORREL_ID	128	X'00000080'
MQPMO_PASS_IDENTITY_CONTEXT	256	X'00000100'
MQPMO_PASS_ALL_CONTEXT	512	X'00000200'
MQPMO_SET_IDENTITY_CONTEXT	1024	X'00000400'
MQPMO_SET_ALL_CONTEXT	2048	X'00000800'
MQPMO_ALTERNATE_USER_AUTHORITY	4096	X'00001000'
MQPMO_FAIL_IF QUIESCING	8192	X'00002000'
MQPMO_NO_CONTEXT	16384	X'00004000'
MQPMO_LOGICAL_ORDER	32768	X'00008000'
MQPMO_ASYNC_RESPONSE	65536	X'00010000'
MQPMO_SYNC_RESPONSE	131072	X'00020000'
MQPMO_RESOLVE_LOCAL_Q	262144	X'00040000'
MQPMO_RETAIN	2097152	X'00200000'
MQPMO_MD_FOR_OUTPUT_ONLY	8388608	X'00800000'
MQPMO_SCOPE_QMGR	67108864	X'04000000'
MQPMO_SUPPRESS_REPLYTO	134217728	X'08000000'
MQPMO_NOT_OWN_SUBS	268435456	X'10000000'
MQPMO_RESPONSE_AS_Q_DEF	0	X'00000000'
MQPMO_RESPONSE_AS_TOPIC_DEF	0	X'00000000'
MQPMO_NONE	0	X'00000000'

### Opções de Colocar Mensagem para máscara de publicação

*Tabela 268. Valores de constantes*

<b>Nome</b>	<b>Valor decimal</b>	<b>Valor Hexadecimal</b>
MQPMO_PUB_OPTIONS_MASK	2097152	X'00200000'

### MQPMRF\_\* (Colocar Campos de Registro de Mensagem)

*Tabela 269. Valores de constantes*

<b>Nome</b>	<b>Valor decimal</b>	<b>Valor Hexadecimal</b>
MQPMRF_MSG_ID	1	X'00000001'
MQPMRF_CORREL_ID	2	X'00000002'
MQPMRF_GROUP_ID	4	X'00000004'
MQPMRF_FEEDBACK	8	X'00000008'
MQPMRF_ACCOUNTING_TOKEN	16	X'00000010'
MQPMRF_NONE	0	X'00000000'

## MQPO\_\* (Opções de limpeza do formato de comando)

Tabela 270. Valores de constantes		
Nome	Valor decimal	Valor Hexadecimal
MQPO_YES	1	X'00000001'
MQPO_NO	0	X'00000000'

## MQPRI\_\* (Prioridade)

Tabela 271. Valores de constantes		
Nome	Valor decimal	Valor Hexadecimal
MQPRI_PRIORITY_AS_Q_DEF	-1	X'FFFFFFFF'
MQPRI_PRIORITY_AS_PARENT	-2	X'FFFFFFFE'
MQPRI_PRIORITY_AS_PUBLISHED	-3	X'FFFFFFFD'
MQPRI_PRIORITY_AS_TOPIC_DEF	-1	X'FFFFFFFF'

## MQPROP\_\* (Valores de Controle de Propriedade de Fila e Canal e Comprimento Máximo de Propriedades)

### Valores de Controle de Propriedade de Fila e Canal

Tabela 272. Valores de constantes		
Nome	Valor decimal	Valor Hexadecimal
MQPROP_COMPATIBILITY	0	X'00000000'
MQPROP_NONE	1	X'00000001'
MQPROP_ALL	2	X'00000002'
MQPROP_FORCE_MQRFH2	3	X'00000003'

### Comprimento máximo das propriedades

Tabela 273. Valores de constantes		
Nome	Valor decimal	Valor Hexadecimal
MQPROP_UNRESTRICTED_LENGTH	-1	X'FFFFFFFF'

## MQPRT\_\* (Valores de resposta de colocação)

Tabela 274. Valores de constantes		
Nome	Valor decimal	Valor Hexadecimal
MQPRT_RESPONSE_AS_PARENT	0	X'00000000'
MQPRT_SYNC_RESPONSE	1	X'00000001'
MQPRT_ASYNC_RESPONSE	2	X'00000002'

## MQPS\_\* (Publicação/Assinatura)

### Status de Publicação / Assinatura do Formato de Comando..

*Tabela 275. Valores de constantes*

Nome	Valor decimal	Valor Hexadecimal
MQPS_STATUS_INACTIVE	0	X'00000000'
MQPS_STATUS_STARTING	1	X'00000001'
MQPS_STATUS_STOPPING	2	X'00000002'
MQPS_STATUS_ACTIVE	3	X'00000003'
MQPS_STATUS_COMPAT	4	X'00000004'
MQPS_STATUS_ERROR	5	X'00000005'
MQPS_STATUS_RECUSADO	6	X'00000006'

### Tags de Publicação / Assinatura como sequências

MQPS_COMMAND	"MQPSCommand"
MQPS_COMP_CODE	"MQPSCompCode"
MQPS_CORREL_ID	"MQPSCorreId"
MQPS_DELETE_OPTIONS	"MQPSDelOpts"
MQPS_ERROR_ID	"MQPSErrorId"
MQPS_ERROR_POS	"MQPSErrorPos"
MQPS_INTEGER_DATA	"MQPSIntData"
MQPS_PARAMETER_ID	"MQSParmId"
MQPS_PUBLICATION_OPTIONS	"MQSPubOpts"
MQPS_PUBLISH_TIMESTAMP	"MQSPubTime"
MQPS_Q_MGR_NAME	"MQPSQMgrName"
MQPS_Q_NAME	"MQPSQName"
MQPS_REASON	"MQPSReason"
MQPS_REASON_TEXT	"MQPSReasonText"
MQPS_REGISTRATION_OPTIONS	"MQPSRegOpts"
MQPS_SEQUENCE_NUMBER	"MQPSeqNum"
MQPS_STREAM_NAME	"MQPSStreamName"
MQPS_STRING_DATA	"MQPSStringData"
MQPS_SUBSCRIPTION_IDENTITY	"MQPSSubIdentity"
MQPS_SUBSCRIPTION_NAME	"MQPSSubName"
MQPS_SUBSCRIPTION_USER_DATA	"MQPSSubUserData"
MQPS_TOPIC	"MQPSTopic"
MQPS_USER_ID	"MQPSUserId"

## Tags de Publicação / Assinatura como sequências em branco

MQPS_COMMAND_B	"-MQPSCommand-
MQPS_COMP_CODE_B	"-MQPSCompCode-
MQPS_CORREL_ID_B	"-MQPSCorrelId-
MQPS_DELETE_OPTIONS_B	"-MQPSDelOpts-
MQPS_ERROR_ID_B	"-MQPSErrorId-
MQPS_ERROR_POS_B	"-MQPSErrorPos-
MQPS_INTEGER_DATA_B	"-MQPSIntData-
MQPS_PARAMETER_ID_B	"-MQPSParmId-
MQPS_PUBLICATION_OPTIONS_B	"-MQPSPubOpts-
MQPS_PUBLISH_TIMESTAMP_B	"-MQPSPubTime-
MQPS_Q_MGR_NAME_B	"-MQPSQMgrName-
MQPS_Q_NAME_B	"-MQPSQName-
MQPS_REASON_B	"-MQPSReason-
MQPS_REASON_TEXT_B	"-MQPSReasonText-
MQPS_REGISTRATION_OPTIONS_B	"-MQPSRegOpts-
MQPS_SEQUENCE_NUMBER_B	"-MQPSSeqNum-
MQPS_STREAM_NAME_B	"-MQPSStreamName-
MQPS_STRING_DATA_B	"-MQPSStringData-
MQPS_SUBSCRIPTION_IDENTITY_B	"-MQPSSubIdentity-
MQPS_SUBSCRIPTION_NAME_B	"-MQPSSubName-
MQPS_SUBSCRIPTION_USER_DATA_B	"-MQPSSubUserData-
MQPS_TOPIC_B	"-MQPSTopic-
MQPS_USER_ID_B	"-MQPSUserId-

**Nota:** O símbolo - representa um único caractere em branco.

## Valores da Tag de Comando de Publicação e Assinatura como sequências.

MQPS_DELETE_PUBLICATION	"DeletePub"
MQPS_DEREGISTER_PUBLISHER	"DeregPub"
MQPS_DEREGISTER_ASSINANTE	"DeregSub"
MQPS_PUBLISH	"Publish"
MQPS_REGISTER_PUBLISHER	"RegPub"
MQPS_REGISTER_ASSINANTE	"RegSub"
MQPS_REQUEST_UPDATE	"ReqUpdate"



## Valores da Tag de Comando de Publicação / Assinatura como sequências em branco

MQPS_DELETE_PUBLICATION_B	"-DeletePub-
MQPS_DEREGISTER_PUBLISHER_B	"-DeregPub-
MQPS_DEREGISTER_SUBSCRIBER_B	"-DeregSub-
MQPS_PUBLISH_B	"-Publish-
MQPS_REGISTER_PUBLISHER_B	"-RegPub-
MQPS_REGISTER_SUBSCRIBER_B	"-RegSub-
MQPS_REQUEST_UPDATE_B	"-ReqUpdate-

**Nota:** O símbolo - representa um único caractere em branco.

## Valores de Tag de Opções de Publicação / Assinatura como sequências.

MQPS_ADD_NAME	"AddName"
MQPS_ANONYMOUS	"Anon"
MQPS_CORREL_ID_AS_IDENTITY	"CorrelAsId"
MQPS_DEREGISTER_ALL	"DeregAll"
MQPS_DIRECT_REQUESTS	"DirectReq"
MQPS_DUPLICATES_OK	"DupsOK"
MQPS_FULL_RESPONSE	"FullResp"
MQPS_INCLUDE_STREAM_NAME	"InclStreamName"
MQPS_INFORM_IF_RETIDO	"InformIfRet"
MQPS_IS_RETAINED_PUBLICATION	"IsRetainedPub"
MQPS_JOIN_EXCLUSIVE	"JoinExcl"
MQPS_JOIN_SHARED	"JoinShared"
MQPS_LEAVE_ONLY	"LeaveOnly"
MQPS_LOCAL	"Local"
MQPS_LOCKED	"Locked"
MQPS_NEW_PUBLICATIONS_ONLY	"NewPubsOnly"
MQPS_NO_ALTERATION	"NoAlter"
MQPS_NO_REGISTRATION	"NoReg"
MQPS_NON_PERSISTENT	"NonPers"
MQPS_NONE	"None"
MQPS_OTHER_SUBSCRIBERS_ONLY	"OtherSubsOnly"
MQPS_PERSISTENT	"Pers"
MQPS_PERSISTENT_AS_PUBLISH	"PersAsPub"
MQPS_PERSISTENT_AS_Q	"PersAsQueue"
MQPS_PUBLISH_ON_REQUEST_ONLY	"PubOnReqOnly"

MQPS_RETAIN_PUBLICATION	"RetainPub"
MQPS_VARIABLE_USER_ID	"VariableUserId"

### Valores de Tag de Opções de Publicação / Assinatura como sequências em branco

MQPS_ADD_NAME_B	"¬AddName¬"
MQPS_ANONYMOUS_B	"¬Anon¬"
MQPS_CORREL_ID_AS_IDENTITY_B	"¬CorrelAsId¬"
MQPS_DEREGISTER_ALL_B	"¬DeregAll¬"
MQPS_DIRECT_REQUESTS_B	"¬DirectReq¬"
MQPS_DUPLICATES_OK_B	"¬DupsOK¬"
MQPS_FULL_RESPONSE_B	"¬FullResp¬"
MQPS_INCLUDE_STREAM_NAME_B	"¬InclStreamName¬"
MQPS_INF_IF_RETAINED_B	"¬InformIfRet¬"
MQPS_IS_RETAINED_PUBLICATION_B	"¬IsRetainedPub¬"
MQPS_JOIN_EXCLUSIVE_B	"¬JoinExcl¬"
MQPS_JOIN_SHARED_B	"¬JoinShared¬"
MQPS_LEAVE_ONLY_B	"¬LeaveOnly¬"
MQPS_LOCAL_B	"¬Local¬"
MQPS_LOCKED_B	"¬Locked¬"
MQPS_NEW_PUBLICATIONS_ONLY_B	"¬NewPubsOnly¬"
MQPS_NO_ALTERATION_B	"¬NoAlter¬"
MQPS_NO_REGISTRATION_B	"¬NoReg¬"
MQPS_NON_PERSISTENT_B	"¬NonPers¬"
MQPS_NONE_B	"¬None¬"
MQPS_OTHER_SUBSCRIBERS_ONLY_B	"¬OtherSubsOnly¬"
MQPS_PERSISTENT_B	"¬Pers¬"
MQPS_PERSISTENT_AS_PUBLISH_B	"¬PersAsPub¬"
MQPS_PERSISTENT_AS_Q_B	"¬PersAsQueue¬"
MQPS_PUBLISH_ON_REQUEST_ONLY_B	"¬PubOnReqOnly¬"
MQPS_RETAIN_PUBLICATION_B	"¬RetainPub¬"
MQPS_VARIABLE_USER_ID_B	"¬VariableUserId¬"

**Nota:** O símbolo ¬ representa um único caractere em branco.

## MQPSC\_\* (Tags de Pasta de Comando de Publicação / Assinatura de Opções de Publicação / Assinatura (psc))

Tabela 276. Valores de constantes		
Nome	Valor decimal	Valor Hexadecimal
MQPSC_FOLDER_VERSION	1	X'00000001'

### MQPSC\_\* (Nomes de Tag de Opções de Publicação / Assinatura)

MQPSC_COMMAND	"Command"
MQPSC_REGISTRATION_OPTION	"RegOpt"
MQPSC_PUBLICATION_OPTION	"PubOpt"
MQPSC_DELETE_OPTION	"DelOpt"
MQPSC_TOPIC	"Topic"
MQPSC_SUBSCRIPTION_POINT	"SubPoint"
MQPSC_FILTER	"Filter"
MQPSC_Q_MGR_NAME	"QMgrName"
MQPSC_Q_NAME	"QName"
MQPSC_PUBLISH_TIMESTAMP	"PubTime"
MQPSC_SEQUENCE_NUMBER	"SeqNum"
MQPSC_SUBSCRIPTION_NAME	"SubName"
MQPSC_SUBSCRIPTION_IDENTITY	"SubIdentity"
MQPSC_SUBSCRIPTION_USER_DATA	"SubUserData"
MQPSC_CORREL_ID	"CorrelId"

### MQPSC\_\* (Nomes de tags XML de tag de publicação / assinatura)

MQPSC_COMMAND_B	"<Command>"
MQPSC_COMMAND_E	"</Command>"
MQPSC_REGISTRATION_OPTION_B	"<RegOpt>"
MQPSC_REGISTRATION_OPTION_E	"</RegOpt>"
MQPSC_PUBLICATION_OPTION_B	"<PubOpt>"
MQPSC_PUBLICATION_OPTION_E	"</PubOpt>"
MQPSC_DELETE_OPTION_B	"<DelOpt>"
MQPSC_DELETE_OPTION_E	"</DelOpt>"
MQPSC_TOPIC_B	"<Topic>"
MQPSC_TOPIC_E	"</Topic>"
MQPSC_SUBSCRIPTION_POINT_B	"<SubPoint>"
MQPSC_SUBSCRIPTION_POINT_E	"</SubPoint>"
MQPSC_FILTER_B	"<Filter>"
MQPSC_FILTER_E	"</Filter>"

MQPSC_Q_MGR_NAME_B	"<QMgrName>"
MQPSC_Q_MGR_NAME_E	"</QMgrName>"
MQPSC_Q_NAME_B	"<QName>"
MQPSC_Q_NAME_E	"</QName>"
MQPSC_PUBLISH_TIMESTAMP_B	"<PubTime>"
MQPSC_PUBLISH_TIMESTAMP_E	"</PubTime>"
MQPSC_SEQUENCE_NUMBER_B	"<SeqNum>"
MQPSC_SEQUENCE_NUMBER_E	"</SeqNum>"
MQPSC_SUBSCRIPTION_NAME_B	"<SubName>"
MQPSC_SUBSCRIPTION_NAME_E	"</SubName>"
MQPSC_SUBSCRIPTION_IDENTITY_B	"<SubIdentity>"
MQPSC_SUBSCRIPTION_IDENTITY_E	"</SubIdentity>"
MQPSC_SUBSCRIPTION_USER_DATA_B	"<SubUserData>"
MQPSC_SUBSCRIPTION_USER_DATA_E	"</SubUserData>"
MQPSC_CORREL_ID_B	"<CorrelId>"
MQPSC_CORREL_ID_E	"</CorrelId>"

**MQPSC\_\* (Valores do Publicador de Tag de Opções de Publicação / Assinatura como seqüências)**

MQPSC_DELETE_PUBLICATION	"DeletePub"
MQPSC_DEREGISTER_ASSINANTE	"DeregSub"
MQPSC_PUBLISH	"Publish"
MQPSC_REGISTER_ASSINANTE	"RegSub"
MQPSC_REQUEST_UPDATE	"ReqUpdate"

**MQPSC\_\* (Valores de nome de tag de opções de publicação / assinatura como seqüências).**

MQPSC_ADD_NAME	"AddName"
MQPSC_CORREL_ID_AS_IDENTITY	"CorrelAsId"
MQPSC_DEREGISTER_ALL	"DeregAll"
MQPSC_DUPLICATES_OK	"DupsOK"
MQPSC_FULL_RESPONSE	"FullResp"
MQPSC_INFORM_IF_RETIDO	"InformIfRet"
MQPSC_IS_RETAINED_PUB	"IsRetainedPub"
MQPSC_JOIN_SHARED	"JoinShared"
MQPSC_JOIN_EXCLUSIVE	"JoinExcl"
MQPSC_LEAVE_ONLY	"LeaveOnly"
MQPSC_LOCAL	"Local"

MQPSC_LOCKED	"Locked"
MQPSC_NEW_PUBS_ONLY	"NewPubsOnly"
MQPSC_NO_ALTERATION	"NoAlter"
MQPSC_NON_PERSISTENT	"NonPers"
MQPSC_OTHER_SUBS_ONLY	"OtherSubsOnly"
MQPSC_PERSISTENT	"Pers"
MQPSC_PERSISTENT_AS_PUBLISH	"PersAsPub"
MQPSC_PERSISTENT_AS_Q	"PersAsQueue"
MQPSC_NONE	"None"
MQPSC_PUB_ON_REQUEST_ONLY	"PubOnReqOnly"
MQPSC_RETAIN_PUB	"RetainPub"
MQPSC_VARIABLE_USER_ID	"VariableUserId"

### MQPSCR\_\* (Opções de Publicação / Assinatura)

#### Opções de Publicação / Assinatura Tags Pasta de Resposta de Publicação / Assinatura (pscr) Tags

*Tabela 277. Valores de constantes*

Nome	Valor decimal	Valor Hexadecimal
MQPSCR_FOLDER_VERSION	1	X'00000001'

#### Nomes de Tag de Opções de Publicação / Assinatura

MQPSCR_CONCLUSÃO	"Completion"
MQPSCR_RESPONSE	"Response"
MQPSCR_REASON	"Reason"

#### Nomes de tags XML de tag de opções de publicação / assinatura

MQPSCR_COMPLETION_B	"<Completion>"
MQPSCR_COMPLETION_E	"</Completion>"
MQPSCR_RESPONSE_B	"<Response>"
MQPSCR_RESPONSE_E	"</Response>"
MQPSCR_REASON_B	"<Reason>"
MQPSCR_REASON_E	"</Reason>"

#### Valores da Tag de Opções de Publicação / Assinatura

MQPSCR_OK	"ok"
MQPSCR_WARNING	"warning"
MQPSCR_ERROR	"error"

## MQPSM\_\* (Modo de Publicação / Assinatura)

Tabela 278. Valores de constantes		
Nome	Valor decimal	Valor Hexadecimal
MQPSM_DISABLED	0	X'00000000'
MQPSM_COMPAT	1	X'00000001'
MQPSM_ENABLED	2	X'00000002'

## MQPSPROP\_\* (Propriedades da Mensagem de Pub / Sub)

Tabela 279. Valores de constantes		
Nome	Valor decimal	Valor Hexadecimal
MQPSPROP_NONE	0	X'00000000'
MQPSPROP_COMPAT	1	X'00000001'
MQPSPROP_RFH2	2	X'00000002'
MQPSPROP_MSGPROP	3	X'00000003'

## MQPSST\_\* (Tipo de status de Pub / Sub do formato de comando)

Tabela 280. Valores de constantes		
Nome	Valor decimal	Valor Hexadecimal
MQPSST_ALL	0	X'00000000'
MQPSST_LOCAL	1	X'00000001'
MQPSST_PARENT	2	X'00000002'
MQPSST_CHILD	3	X'00000003'

## MQPUBO\_\* (Opções de Publicação de Publicação / Assinatura)

Tabela 281. Valores de constantes		
Nome	Valor decimal	Valor Hexadecimal
MQPUBO_NONE	0	X'00000000'
MQPUBO_CORREL_ID_AS_IDENTITY	1	X'00000001'
MQPUBO_RETAIN_PUBLICATION	2	X'00000002'
MQPUBO_OTHER_SUBSCRIBERS_ONLY	4	X'00000004'
MQPUBO_NO_REGISTRATION	8	X'00000008'
MQPUBO_IS_RETAINED_PUBLICATION	16	X'00000010'

## MQPXP\_\* (estrutura de parâmetro de saída de roteamento de publicação / assinatura)

Tabela 282. Estruturas de constantes	
Nome	Estrutura
MQPXP_STRUC_ID	"PXP~"
MQPXP_STRUC_ID_ARRAY	'P', 'X', 'P', '~'

**Nota:** O símbolo ~ representa um único caractere em branco.

<i>Tabela 283. Valores de constantes</i>		
<b>Nome</b>	<b>Valor decimal</b>	<b>Valor Hexadecimal</b>
MQPXP_VERSION_1	1	X'00000001'
MQPXP_CURRENT_VERSION	1	X'00000001'

## **MQQA\_\* (Atributos de fila).**

### **Inibir valores de obtenção**

<i>Tabela 284. Valores de constantes</i>		
<b>Nome</b>	<b>Valor decimal</b>	<b>Valor Hexadecimal</b>
MQQA_GET_INHIBITED	1	X'00000001'
MQQA_GET_ALLOWED	0	X'00000000'

### **Inibir valores de colocação**

<i>Tabela 285. Valores de constantes</i>		
<b>Nome</b>	<b>Valor decimal</b>	<b>Valor Hexadecimal</b>
MQQA_PUT_INHIBITED	1	X'00000001'
MQQA_PUT_ALLOWED	0	X'00000000'

### **Compartilhamento de Fila**

<i>Tabela 286. Valores de constantes</i>		
<b>Nome</b>	<b>Valor decimal</b>	<b>Valor Hexadecimal</b>
MQQA_SHAREABLE	1	X'00000001'
MQQA_NOT_SHAREABLE	0	X'00000000'

### **Reforço de Reverso**

<i>Tabela 287. Valores de constantes</i>		
<b>Nome</b>	<b>Valor decimal</b>	<b>Valor Hexadecimal</b>
MQQA_BACKOUT_HARDENED	1	X'00000001'
MQQA_BACKOUT_NOT_HARDENIDOS	0	X'00000000'

## **MQQDT\_\* (Tipos de Definição de Fila)**

<i>Tabela 288. Valores de constantes</i>		
<b>Nome</b>	<b>Valor decimal</b>	<b>Valor Hexadecimal</b>
MQQDT_PREDEFINED	1	X'00000001'
MQQDT_PERMANENT_DYNAMIC	2	X'00000002'
MQQDT_TEMPORARY_DYNAMIC	3	X'00000003'
MQQDT_SHARED_DYNAMIC	4	X'00000004'

## MQQF\_\* (Sinalizadores de Fila)

Tabela 289. Valores de constantes		
Nome	Valor decimal	Valor Hexadecimal
MQQF_LOCAL_Q	1	X'00000001'
MQQF_CLWL_USEQ_ANY	64	X'00000040'
MQQF_CLWL_USEQ_LOCAL	128	X'00000080'

## MQQMDT\_\* (Tipos de Definição do Gerenciador de Filas do formato de comando)..

Tabela 290. Valores de constantes		
Nome	Valor decimal	Valor Hexadecimal
MQQMDT_EXPLICIT_CLUSTER_SENDER	1	X'00000001'
MQQMDT_AUTO_CLUSTER_SENDER	2	X'00000002'
MQQMDT_AUTO_EXP_CLUSTER_SENDER	4	X'00000004'
MQQMDT_CLUSTER_RECEIVER	3	X'00000003'

## MQQMF\_\* (Sinalizações do Gerenciador de Filas)

Tabela 291. Valores de constantes		
Nome	Valor decimal	Valor Hexadecimal
MQQMF_REPOSITORY_Q_MGR	2	X'00000002'
MQQMF_CLUSSDR_USER_DEFINED	8	X'00000008'
MQQMF_CLUSSDR_AUTO_DEFINED	16	X'00000010'
MQQMF_AVAILABLE	32	X'00000020'

## MQQMFACT\_\* (Formato de Comando Queue Manager Facility)

Tabela 292. Valores de constantes		
Nome	Valor decimal	Valor Hexadecimal
MQQMFACT_IMS_BRIDGE	1	X'00000001'
MQQMFACT_DB2	2	X'00000002'

## MQQMSTA\_\* (Status do Gerenciador de Filas no formato de comando).

Tabela 293. Valores de constantes		
Nome	Valor decimal	Valor Hexadecimal
MQQMSTA_STARTING	1	X'00000001'
MQQMSTA_RUNNING	2	X'00000002'
MQQMSTA QUIESCING	3	X'00000003'

## MQQMT\_\* (Tipos de Gerenciador de Filas de Formato de Comandos)..

Tabela 294. Valores de constantes		
Nome	Valor decimal	Valor Hexadecimal
MQQMT_NORMAL	0	X'00000000'



<i>Tabela 294. Valores de constantes (continuação)</i>		
<b>Nome</b>	<b>Valor decimal</b>	<b>Valor Hexadecimal</b>
MQQMT_REPOSITORY	1	X'00000001'

### **MQQO\_\* (Opções de quiesce do formato de comando)..**

<i>Tabela 295. Valores de constantes</i>		
<b>Nome</b>	<b>Valor decimal</b>	<b>Valor Hexadecimal</b>
MQQO_YES	1	X'00000001'
MQQO_NO	0	X'00000000'

### **MQQSGD\_\* (Disposições do grupo de filas compartilhadas)**

<i>Tabela 296. Valores de constantes</i>		
<b>Nome</b>	<b>Valor decimal</b>	<b>Valor Hexadecimal</b>
MQQSGD_ALL	-1	X'FFFFFFFF'
MQQSGD_Q_MGR	0	X'00000000'
MQQSGD_COPY	1	X'00000001'
MQQSGD_SHARED	2	X'00000002'
MQQSGD_GROUP	3	X'00000003'
MQQSGD_PRIVATE	4	X'00000004'
MQQSGD_LIVE	6	X'00000006'

### **MQQSGS\_\* (Status do grupo de filas compartilhadas no formato de comando)**

<i>Tabela 297. Valores de constantes</i>		
<b>Nome</b>	<b>Valor decimal</b>	<b>Valor Hexadecimal</b>
MQQSGS_UNKNOWN	0	X'00000000'
MQQSGS_CREATED	1	X'00000001'
MQQSGS_ACTIVE	2	X'00000002'
MQQSGS_INACTIVE	3	X'00000003'
MQQSGS_FAILED	4	X'00000004'
MQQSGS_PENDING	5	X'00000005'

### **MQQSIE\_\* (Serviço de fila de formato de comando-Eventos de intervalo)**

<i>Tabela 298. Valores de constantes</i>		
<b>Nome</b>	<b>Valor decimal</b>	<b>Valor Hexadecimal</b>
MQQSIE_NONE	0	X'00000000'
MQQSIE_HIGH	1	X'00000001'
MQQSIE_OK	2	X'00000002'

## MQQSO\_\* (Opções Abertas de Status da Fila de Formato de Comando para SET, BROWSE, INPUT)

<i>Tabela 299. Valores de constantes</i>		
Nome	Valor decimal	Valor Hexadecimal
MQQSO_NO	0	X'00000000'
MQQSO_YES	1	X'00000001'
MQQSO_SHARED	1	X'00000001'
MQQSO_EXCLUSIVE	2	X'00000002'

## MQQSOT\_\* (Tipos Abertos de Status da Fila de formato de comando)

<i>Tabela 300. Valores de constantes</i>		
Nome	Valor decimal	Valor Hexadecimal
MQQSOT_ALL	1	X'00000001'
MQQSOT_INPUT	2	X'00000002'
MQQSOT_OUTPUT	3	X'00000003'

## MQQSUM\_\* (Mensagens não confirmadas do status da fila de formato de comando).

<i>Tabela 301. Valores de constantes</i>		
Nome	Valor decimal	Valor Hexadecimal
MQQSUM_YES	1	X'00000001'
MQQSUM_NO	0	X'00000000'

## MQQT\_\* (Tipos de Fila e Tipos de Fila Estendidos)

### Tipos de Fila

<i>Tabela 302. Valores de constantes</i>		
Nome	Valor decimal	Valor Hexadecimal
MQQT_LOCAL	1	X'00000001'
MQQT_MODEL	2	X'00000002'
MQQT_ALIAS	3	X'00000003'
MQQT_REMOTE	6	X'00000006'
MQQT_CLUSTER	7	X'00000007'

### Tipos de Fila Estendidos

<i>Tabela 303. Valores de constantes</i>		
Nome	Valor decimal	Valor Hexadecimal
MQQT_ALL	1001	X'000003E9'

## MQRC\_\* (códigos de razão)

*Tabela 304. Valores de constantes*

Nome	Valor decimal	Valor Hexadecimal
MQRC_NONE	0	X'00000000'
MQRC_APPL_FIRST	900	X'00000384'
MQRC_APPL_LAST	999	X'000003E7'
MQRC_ALIAS_BASE_Q_TYPE_ERROR	2001	X'000007D1'
MQRC_ALREADY_CONNECTED	2002	X'000007D2'
MQRC_BACKED_OUT	2003	X'000007D3'
MQRC_BUFFER_ERROR	2004	X'000007D4'
MQRC_BUFFER_LENGTH_ERROR	2005	X'000007D5'
MQRC_CHAR_ATTR_LENGTH_ERROR	2006	X'000007D6'
MQRC_CHAR_ATTRS_ERROR	2007	X'000007D7'
MQRC_CHAR_ATTRS_TOO_SHORT	2008	X'000007D8'
MQRC_CONNECTION_BROKEN	2009	X'000007D9'
MQRC_DATA_LENGTH_ERROR	2010	X'000007DA'
MQRC_DYNAMIC_Q_NAME_ERROR	2011	X'000007DB'
MQRC_ENVIRONMENT_ERROR	2012	X'000007DC'
MQRC_EXPIRY_ERROR	2013	X'000007DD'
MQRC_FEEDBACK_ERROR	2014	X'000007DE'
MQRC_GET_INHIBITED	2016	X'000007E0'
MQRC_HANDLE_NOT_AVAILABLE	2017	X'000007E1'
MQRC_HCONN_ERROR	2018	X'000007E2'
MQRC_HOBJ_ERROR	2019	X'000007E3'
MQRC_INHIBIT_VALUE_ERROR	2020	X'000007E4'
MQRC_INT_ATTR_COUNT_ERROR	2021	X'000007E5'
MQRC_INT_ATTR_COUNT_TOO_SMALL	2022	X'000007E6'
MQRC_INT_ATTRS_ARRAY_ERROR	2023	X'000007E7'
MQRC_SYNCPOINT_LIMIT_REACHED	2024	X'000007E8'
MQRC_MAX_CONNS_LIMIT_REACHED	2025	X'000007E9'
MQRC_MD_ERROR	2026	X'000007EA'
MQRC_MISSING_REPLY_TO_Q	2027	X'000007EB'
MQRC_MSG_TYPE_ERROR	2029	X'000007ED'
MQRC_MSG_TOO_BIG_FOR_Q	2030	X'000007EE'
MQRC_MSG_TOO_BIG_FOR_Q_MGR	2031	X'000007EF'
MQRC_NO_MSG_AVAILABLE	2033	X'000007F1'
MQRC_NO_MSG_UNDER_CURSOR	2034	X'000007F2'
MQRC_NOT_AUTHORIZED	2035	X'000007F3'
MQRC_NOT_OPEN_FOR_BROWSE	2036	X'000007F4'
MQRC_NOT_OPEN_FOR_INPUT	2037	X'000007F5'
MQRC_NOT_OPEN_FOR_INQUIRE	2038	X'000007F6'

Tabela 304. Valores de constantes (continuação)

Nome	Valor decimal	Valor Hexadecimal
MQRC_NOT_OPEN_FOR_OUTPUT	2039	X'000007F7'
MQRC_NOT_OPEN_FOR_SET	2040	X'000007F8'
MQRC_OBJECT_CHANGED	2041	X'000007F9'
MQRC_OBJECT_IN_USE	2042	X'000007FA'
MQRC_OBJECT_TYPE_ERROR	2043	X'000007FB'
MQRC_OD_ERROR	2044	X'000007FC'
MQRC_OPTION_NOT_VALID_FOR_TYPE	2045	X'000007FD'
MQRC_OPTIONS_ERROR	2046	X'000007FE'
MQRC_PERSISTENCE_ERROR	2047	X'000007FF'
MQRC_PERSISTENT_NOT_ALLOWED	2048	X'00000800'
MQRC_PRIORITY_EXCEEDS_MAXIMUM	2049	X'00000801'
MQRC_PRIORITY_ERROR	2050	X'00000802'
MQRC_PUT_INHIBITED	2051	X'00000803'
MQRC_Q_DELETED	2052	X'00000804'
MQRC_Q_FULL	2053	X'00000805'
MQRC_Q_NOT_EMPTY	2055	X'00000807'
MQRC_Q_SPACE_NOT_AVAILABLE	2056	X'00000808'
MQRC_Q_TYPE_ERROR	2057	X'00000809'
MQRC_Q_MGR_NAME_ERROR	2058	X'0000080A'
MQRC_Q_MGR_NOT_AVAILABLE	2059	X'0000080B'
MQRC_REPORT_OPTIONS_ERROR	2061	X'0000080D'
MQRC_SECOND_MARK_NOT_ALLOWED	2062	X'0000080E'
MQRC_SECURITY_ERROR	2063	X'0000080F'
MQRC_SELECTOR_COUNT_ERROR	2065	X'00000811'
MQRC_SELECTOR_LIMIT_EXCEDIDO	2066	X'00000812'
MQRC_SELECTOR_ERROR	2067	X'00000813'
MQRC_SELECTOR_NOT_FOR_TYPE	2068	X'00000814'
MQRC_SIGNAL_OUTSTANDING	2069	X'00000815'
MQRC_SIGNAL_REQUEST_ACCEPTED	2070	X'00000816'
MQRC_STORAGE_NOT_AVAILABLE	2071	X'00000817'
MQRC_SYNCPOINT_NOT_AVAILABLE	2072	X'00000818'
MQRC_TRIGGER_CONTROL_ERROR	2075	X'0000081B'
MQRC_TRIGGER_DEPTH_ERROR	2076	X'0000081C'
MQRC_TRIGGER_MSG_PRIORITY_ERR	2077	X'0000081D'
MQRC_TRIGGER_TYPE_ERROR	2078	X'0000081E'
MQRC_TRUNCATED_MSG_ACCEPTED	2079	X'0000081F'
MQRC_TRUNCATED_MSG_FAILED	2080	X'00000820'
MQRC_UNKNOWN_ALIAS_BASE_Q	2082	X'00000822'
MQRC_UNKNOWN_OBJECT_NAME	2085	X'00000825'

Tabela 304. Valores de constantes (continuação)

Nome	Valor decimal	Valor Hexadecimal
MQRC_UNKNOWN_OBJECT_Q_MGR	2086	X'00000826'
MQRC_UNKNOWN_REMOTE_Q_MGR	2087	X'00000827'
MQRC_WAIT_INTERVAL_ERROR	2090	X'0000082A'
MQRC_XMIT_Q_TYPE_ERROR	2091	X'0000082B'
MQRC_XMIT_Q_USAGE_ERROR	2092	X'0000082C'
MQRC_NOT_OPEN_FOR_PASS_ALL	2093	X'0000082D'
MQRC_NOT_OPEN_FOR_PASS_IDENT	2094	X'0000082E'
MQRC_NOT_OPEN_FOR_SET_ALL	2095	X'0000082F'
MQRC_NOT_OPEN_FOR_SET_IDENT	2096	X'00000830'
MQRC_CONTEXT_HANDLE_ERROR	2097	X'00000831'
MQRC_CONTEXT_NOT_AVAILABLE	2098	X'00000832'
MQRC_SIGNAL1_ERROR	2099	X'00000833'
MQRC_OBJECT_ALREADY_EXISTS	2100	X'00000834'
MQRC_OBJECT_DAMAGED	2101	X'00000835'
MQRC_RESOURCE_PROBLEM	2102	X'00000836'
MQRC_ANOTHER_Q_MGR_CONNECTED	2103	X'00000837'
MQRC_UNKNOWN_REPORT_OPTION	2104	X'00000838'
MQRC_STORAGE_CLASS_ERROR	2105	X'00000839'
MQRC_COD_NOT_VALID_FOR_XCF_Q	2106	X'0000083A'
MQRC_XWAIT_CANCELED	2107	X'0000083B'
MQRC_XWAIT_ERROR	2108	X'0000083C'
MQRC_SUPPRESSED_BY_EXIT	2109	X'0000083D'
MQRC_FORMAT_ERROR	2110	X'0000083E'
MQRC_SOURCE_CCSD_ERROR	2111	X'0000083F'
MQRC_SOURCE_INTEGER_ENC_ERROR	2112	X'00000840'
MQRC_SOURCE_DECIMAL_ENC_ERROR	2113	X'00000841'
MQRC_SOURCE_FLOAT_ENC_ERROR	2114	X'00000842'
MQRC_TARGET_CCSD_ERROR	2115	X'00000843'
MQRC_TARGET_INTEGER_ENC_ERROR	2116	X'00000844'
MQRC_TARGET_DECIMAL_ENC_ERROR	2117	X'00000845'
MQRC_TARGET_FLOAT_ENC_ERROR	2118	X'00000846'
MQRC_NOT_CONVERTED	2119	X'00000847'
MQRC_CONVERTED_MSG_TOO_BIG	2120	X'00000848'
MQRC_TRUNCATED	2120	X'00000848'
MQRC_NO_EXTERNAL_PARTICIPANTES	2121	X'00000849'
MQRC_PARTICIPANT_NOT_AVAILABLE	2122	X'0000084A'
MQRC_OUTCOME_MIXED	2123	X'0000084B'
MQRC_OUTCOME_PENDING	2124	X'0000084C'
MQRC_BRIDGE_STARTED	2125	X'0000084D'

Tabela 304. Valores de constantes (continuação)

Nome	Valor decimal	Valor Hexadecimal
MQRC_BRIDGE_STOPPED	2126	X'0000084E'
MQRC_ADAPTER_STORAGE_SHORTAGE	2127	X'0000084F'
MQRC_UOW_IN_PROGRESS	2128	X'00000850'
MQRC_ADAPTER_CONN_LOAD_ERROR	2129	X'00000851'
MQRC_ADAPTER_SERV_LOAD_ERROR	2130	X'00000852'
MQRC_ADAPTER_DEFS_ERROR	2131	X'00000853'
MQRC_ADAPTER_DEFS_LOAD_ERROR	2132	X'00000854'
MQRC_ADAPTER_CONV_LOAD_ERROR	2133	X'00000855'
MQRC_BO_ERROR	2134	X'00000856'
MQRC_DH_ERROR	2135	X'00000857'
MQRC_MULTIPLE_REASONS	2136	X'00000858'
MQRC_OPEN_FAILED	2137	X'00000859'
MQRC_ADAPTER_DISC_LOAD_ERROR	2138	X'0000085A'
MQRC_CNO_ERROR	2139	X'0000085B'
MQRC_CICS_WAIT_FAILED	2140	X'0000085C'
MQRC_DLH_ERROR	2141	X'0000085D'
MQRC_HEADER_ERROR	2142	X'0000085E'
MQRC_SOURCE_LENGTH_ERROR	2143	X'0000085F'
MQRC_TARGET_LENGTH_ERROR	2144	X'00000860'
MQRC_SOURCE_BUFFER_ERROR	2145	X'00000861'
MQRC_TARGET_BUFFER_ERROR	2146	X'00000862'
MQRC_IIH_ERROR	2148	X'00000864'
MQRC_PCF_ERROR	2149	X'00000865'
MQRC_DBCS_ERROR	2150	X'00000866'
MQRC_OBJECT_NAME_ERROR	2152	X'00000868'
MQRC_OBJECT_Q_MGR_NAME_ERROR	2153	X'00000869'
MQRC_RECS_PRESENT_ERROR	2154	X'0000086A'
MQRC_OBJECT_RECORDS_ERROR	2155	X'0000086B'
MQRC_RESPONSE_RECORDS_ERROR	2156	X'0000086C'
MQRC_ASID_MISMATCH	2157	X'0000086D'
MQRC_PMO_RECORD_FLAGS_ERROR	2158	X'0000086E'
MQRC_PUT_MSG_RECORDS_ERROR	2159	X'0000086F'
MQRC_CONN_ID_IN_USE	2160	X'00000870'
MQRC_Q_MGR QUIESCING	2161	X'00000871'
MQRC_Q_MGR_STOPPING	2162	X'00000872'
MQRC_DUPLICATE_RECOV_COORD	2163	X'00000873'
MQRC_PMO_ERROR	2173	X'0000087D'
MQRC_API_EXIT_NOT_FOUND	2182	X'00000886'
MQRC_API_EXIT_LOAD_ERROR	2183	X'00000887'

Tabela 304. Valores de constantes (continuação)

Nome	Valor decimal	Valor Hexadecimal
MQRC_REMOTE_Q_NAME_ERROR	2184	X'00000888'
MQRC_INCONSISTENT_PERSISTENCE	2185	X'00000889'
MQRC_GMO_ERROR	2186	X'0000088A'
MQRC_CICS_BRIDGE_RESTRICTION	2187	X'0000088B'
MQRC_STOPPED_BY_CLUSTER_EXIT	2188	X'0000088C'
MQRC_CLUSTER_RESOLUTION_ERROR	2189	X'0000088D'
MQRC_CONVERTED_STRING_TOO_BIG	2190	X'0000088E'
MQRC_TMC_ERROR	2191	X'0000088F'
MQRC_PAGESET_FULL	2192	X'00000890'
MQRC_STORAGE_MEDIUM_FULL	2192	X'00000890'
MQRC_PAGESET_ERROR	2193	X'00000891'
MQRC_NAME_NOT_VALID_FOR_TYPE	2194	X'00000892'
MQRC_UNEXPECTED_ERROR	2195	X'00000893'
MQRC_UNKNOWN_XMIT_Q	2196	X'00000894'
MQRC_UNKNOWN_DEF_XMIT_Q	2197	X'00000895'
MQRC_DEF_XMIT_Q_TYPE_ERROR	2198	X'00000896'
MQRC_DEF_XMIT_Q_USAGE_ERROR	2199	X'00000897'
MQRC_MSG_MARKED_BROWSE_CO_OP	2200	X'00000898'
MQRC_NAME_IN_USE	2201	X'00000899'
MQRC_CONNECTION QUIESCING	2202	X'0000089A'
MQRC_CONNECTION_STOPPING	2203	X'0000089B'
MQRC_ADAPTER_NOT_AVAILABLE	2204	X'0000089C'
MQRC_MSG_ID_ERROR	2206	X'0000089E'
MQRC_CORREL_ID_ERROR	2207	X'0000089F'
MQRC_FILE_SYSTEM_ERROR	2208	X'000008A0'
MQRC_NO_MSG_LOCKED	2209	X'000008A1'
MQRC_SOAP_DOTNET_ERROR	2210	X'000008A2'
MQRC_SOAP_AXIS_ERROR	2211	X'000008A3'
MQRC_SOAP_URL_ERROR	2212	X'000008A4'
MQRC_FILE_NOT_AUDITED	2216	X'000008A8'
MQRC_CONNECTION_NOT_AUTHORIZED	2217	X'000008A9'
MQRC_MSG_TOO_BIG_FOR_CHANEXOL	2218	X'000008AA'
MQRC_CALL_IN_PROGRESS	2219	X'000008AB'
MQRC_RMH_ERROR	2220	X'000008AC'
MQRC_Q_MGR_ACTIVE	2222	X'000008AE'
MQRC_Q_MGR_NOT_ACTIVE	2223	X'000008AF'
MQRC_Q_DEPTH_HIGH	2224	X'000008B0'
MQRC_Q_DEPTH_LOW	2225	X'000008B1'
MQRC_Q_SERVICE_INTERVAL_HIGH	2226	X'000008B2'

Tabela 304. Valores de constantes (continuação)

Nome	Valor decimal	Valor Hexadecimal
MQRC_Q_SERVICE_INTERVAL_OK	2227	X'000008B3'
MQRC_RFH_HEADER_FIELD_ERROR	2228	X'000008B4'
MQRC_RAS_PROPERTY_ERROR	2229	X'000008B5'
MQRC_UNIT_OF_WORK_NOT_STARTED	2232	X'000008B8'
MQRC_CHANNEL_AUTO_DEF_OK	2233	X'000008B9'
MQRC_CHANNEL_AUTO_DEF_ERROR	2234	X'000008BA'
MQRC_CFH_ERROR	2235	X'000008BB'
MQRC_CFIL_ERROR	2236	X'000008BC'
MQRC_CFIN_ERROR	2237	X'000008BD'
MQRC_CFSL_ERROR	2238	X'000008BE'
MQRC_CFST_ERROR	2239	X'000008BF'
MQRC_INCOMPLETE_GROUP	2241	X'000008C1'
MQRC_INCOMPLETE_MSG	2242	X'000008C2'
MQRC_INCONSISTENT_CCIDS	2243	X'000008C3'
MQRC_INCONSISTENT_ENCODINGS	2244	X'000008C4'
MQRC_INCONSISTENT_UOW	2245	X'000008C5'
MQRC_INVALID_MSG_UNDER_CURSOR	2246	X'000008C6'
MQRC_MATCH_OPTIONS_ERROR	2247	X'000008C7'
MQRC_MDE_ERROR	2248	X'000008C8'
MQRC_MSG_FLAGS_ERROR	2249	X'000008C9'
MQRC_MSG_SEQ_NUMBER_ERROR	2250	X'000008CA'
MQRC_OFFSET_ERROR	2251	X'000008CB'
MQRC_ORIGINAL_LENGTH_ERROR	2252	X'000008CC'
MQRC_SEGMENT_LENGTH_ZERO	2253	X'000008CD'
MQRC_UOW_NOT_AVAILABLE	2255	X'000008CF'
MQRC_WRONG_GMO_VERSION	2256	X'000008D0'
MQRC_WRONG_MD_VERSION	2257	X'000008D1'
MQRC_GROUP_ID_ERROR	2258	X'000008D2'
MQRC_INCONSISTENT_BROWSE	2259	X'000008D3'
MQRC_XQH_ERROR	2260	X'000008D4'
MQRC_SRC_ENV_ERROR	2261	X'000008D5'
MQRC_SRC_NAME_ERROR	2262	X'000008D6'
MQRC_DEST_ENV_ERROR	2263	X'000008D7'
MQRC_DEST_NAME_ERROR	2264	X'000008D8'
MQRC_TM_ERROR	2265	X'000008D9'
MQRC_CLUSTER_EXIT_ERROR	2266	X'000008DA'
MQRC_CLUSTER_EXIT_LOAD_ERROR	2267	X'000008DB'
MQRC_CLUSTER_PUT_INIBIDA	2268	X'000008DC'
MQRC_CLUSTER_RESOURCE_ERROR	2269	X'000008DD'



Tabela 304. Valores de constantes (continuação)

Nome	Valor decimal	Valor Hexadecimal
MQRC_NO_DESTINATIONS_AVAILABLE	2270	X'000008DE'
MQRC_CONN_TAG_IN_USE	2271	X'000008DF'
MQRC_PARTIALLY_CONVERTED	2272	X'000008E0'
MQRC_CONNECTION_ERROR	2273	X'000008E1'
MQRC_OPTION_ENVIRONMENT_ERROR	2274	X'000008E2'
MQRC_CD_ERROR	2277	X'000008E5'
MQRC_CLIENT_CONN_ERROR	2278	X'000008E6'
MQRC_CHANNEL_STOPPED_BY_USER	2279	X'000008E7'
MQRC_HCONFIG_ERROR	2280	X'000008E8'
MQRC_FUNCTION_ERROR	2281	X'000008E9'
MQRC_CHANNEL_STARTED	2282	X'000008EA'
MQRC_CHANNEL_STOPPED	2283	X'000008EB'
MQRC_CHANNEL_CONV_ERROR	2284	X'000008EC'
MQRC_SERVICE_NOT_AVAILABLE	2285	X'000008ED'
MQRC_INITIALIZATION_FAILED	2286	X'000008EE'
MQRC_TERMINATION_FAILED	2287	X'000008EF'
MQRC_UNKNOWN_Q_NAME	2288	X'000008F0'
MQRC_SERVICE_ERROR	2289	X'000008F1'
MQRC_Q_ALREADY_EXISTS	2290	X'000008F2'
MQRC_USER_ID_NOT_AVAILABLE	2291	X'000008F3'
MQRC_UNKNOWN_ENTITY	2292	X'000008F4'
MQRC_UNKNOWN_AUTH_ENTITY	2293	X'000008F5'
MQRC_UNKNOWN_REF_OBJECT	2294	X'000008F6'
MQRC_CHANNEL_ACTIVATED	2295	X'000008F7'
MQRC_CHANNEL_NOT_ACTIVATED	2296	X'000008F8'
MQRC_UOW_CANCELADO	2297	X'000008F9'
MQRC_FUNCTION_NOT_SUPPORTED	2298	X'000008FA'
MQRC_SELECTOR_TYPE_ERROR	2299	X'000008FB'
MQRC_COMMAND_TYPE_ERROR	2300	X'000008FC'
MQRC_MULTIPLE_INSTANCE_ERROR	2301	X'000008FD'
MQRC_SYSTEM_ITEM_NOT_ALTERABLE	2302	X'000008FE'
MQRC_BAG_CONVERSION_ERROR	2303	X'000008FF'
MQRC_SELECTOR_OUT_OF_RANGE	2304	X'00000900'
MQRC_SELECTOR_NOT_UNIQUE	2305	X'00000901'
MQRC_INDEX_NOT_PRESENT	2306	X'00000902'
MQRC_STRING_ERROR	2307	X'00000903'
MQRC_ENCODING_NOT_SUPPORTED	2308	X'00000904'
MQRC_SELECTOR_NOT_PRESENT	2309	X'00000905'
MQRC_OUT_SELECTOR_ERROR	2310	X'00000906'

Tabela 304. Valores de constantes (continuação)

Nome	Valor decimal	Valor Hexadecimal
MQRC_STRING_TRUNCATED	2311	X'00000907'
MQRC_SELECTOR_ERRONE_TYPE	2312	X'00000908'
MQRC_INCONSISTENT_ITEM_TYPE	2313	X'00000909'
MQRC_INDEX_ERROR	2314	X'0000090A'
MQRC_SYSTEM_BAG_NOT_ALTERABLE	2315	X'0000090B'
MQRC_ITEM_COUNT_ERROR	2316	X'0000090C'
MQRC_FORMAT_NOT_SUPPORTED	2317	X'0000090D'
MQRC_SELECTOR_NOT_SUPPORTED	2318	X'0000090E'
MQRC_ITEM_VALUE_ERROR	2319	X'0000090F'
MQRC_HBAG_ERROR	2320	X'00000910'
MQRC_PARAMETER_MISSING	2321	X'00000911'
MQRC_CMD_SERVER_NOT_AVAILABLE	2322	X'00000912'
MQRC_STRING_LENGTH_ERROR	2323	X'00000913'
MQRC_INQUIRY_COMMAND_ERROR	2324	X'00000914'
MQRC_NESTED_BAG_NOT_SUPPORTED	2325	X'00000915'
MQRC_BAG_ERR_TYPE	2326	X'00000916'
MQRC_ITEM_TYPE_ERROR	2327	X'00000917'
MQRC_SYSTEM_BAG_NOT_DELETABLE	2328	X'00000918'
MQRC_SYSTEM_ITEM_NOT_DELETABLE	2329	X'00000919'
MQRC_CODED_CHAR_SET_ID_ERROR	2330	X'0000091A'
MQRC_MSG_TOKEN_ERROR	2331	X'0000091B'
MQRC_MISSING_WIH	2332	X'0000091C'
MQRC_WIH_ERROR	2333	X'0000091D'
MQRC_RFH_ERROR	2334	X'0000091E'
MQRC_RFH_STRING_ERROR	2335	X'0000091F'
MQRC_RFH_COMMAND_ERROR	2336	X'00000920'
MQRC_RFH_PARM_ERROR	2337	X'00000921'
MQRC_RFH_DUPLICATE_PARM	2338	X'00000922'
MQRC_RFH_PARM_MISSING	2339	X'00000923'
MQRC_CHAR_CONVERSION_ERROR	2340	X'00000924'
MQRC_UCS2_CONVERSION_ERROR	2341	X'00000925'
MQRC_DB2_NOT_AVAILABLE	2342	X'00000926'
MQRC_OBJECT_NOT_UNIQUE	2343	X'00000927'
MQRC_CONN_TAG_NOT_LIBERADO	2344	X'00000928'
MQRC_CF_NOT_AVAILABLE	2345	X'00000929'
MQRC_CF_STRUC_IN_USE	2346	X'0000092A'
MQRC_CF_STRUC_LIST_HDR_IN_USE	2347	X'0000092B'
MQRC_CF_STRUC_AUTH_FAILED	2348	X'0000092C'
MQRC_CF_STRUC_ERROR	2349	X'0000092D'

Tabela 304. Valores de constantes (continuação)

Nome	Valor decimal	Valor Hexadecimal
MQRC_CONN_TAG_NOT_USABLE	2350	X'0000092E'
MQRC_GLOBAL_UOW_CONFLICT	2351	X'0000092F'
MQRC_LOCAL_UOW_CONFLICT	2352	X'00000930'
MQRC_HANDLE_IN_USE_FOR_UOW	2353	X'00000931'
MQRC_UOW_ENLISTMENT_ERROR	2354	X'00000932'
MQRC_UOW_MIX_NOT_SUPPORTED	2355	X'00000933'
MQRC_WXP_ERROR	2356	X'00000934'
MQRC_CURRENT_RECORD_ERROR	2357	X'00000935'
MQRC_NEXT_OFFSET_ERROR	2358	X'00000936'
MQRC_NO_RECORD_AVAILABLE	2359	X'00000937'
MQRC_OBJECT_LEVEL_INCOMPATÍVEL	2360	X'00000938'
MQRC_NEXT_RECORD_ERROR	2361	X'00000939'
MQRC_BACKOUT_THRESHOLD_ALCANÇADA	2362	X'0000093A'
MQRC_MSG_NOT_MATCHED	2363	X'0000093B'
MQRC_JMS_FORMAT_ERROR	2364	X'0000093C'
MQRC_SEGMENTS_NOT_SUPPORTED	2365	X'0000093D'
MQRC_ERRR_CF_LEVEL	2366	X'0000093E'
MQRC_CONFIG_CREATE_OBJECT	2367	X'0000093F'
MQRC_CONFIG_CHANGE_OBJECT	2368	X'00000940'
MQRC_CONFIG_DELETE_OBJECT	2369	X'00000941'
MQRC_CONFIG_REFRESH_OBJECT	2370	X'00000942'
MQRC_CHANNEL_SSL_ERROR	2371	X'00000943'
MQRC_PARTICIPANT_NOT_DEFINED	2372	X'00000944'
MQRC_CF_STRUC_FAILED	2373	X'00000945'
MQRC_API_EXIT_ERROR	2374	X'00000946'
MQRC_API_EXIT_INIT_ERROR	2375	X'00000947'
MQRC_API_EXIT_TERM_ERROR	2376	X'00000948'
MQRC_EXIT_REASON_ERROR	2377	X'00000949'
MQRC_RESERVED_VALUE_ERROR	2378	X'0000094A'
MQRC_NO_DATA_AVAILABLE	2379	X'0000094B'
MQRC_SCO_ERROR	2380	X'0000094C'
MQRC_KEY_REPOSITORY_ERROR	2381	X'0000094D'
MQRC_CRYPT0_HARDWARE_ERROR	2382	X'0000094E'
MQRC_AUTH_INFO_REC_COUNT_ERROR	2383	X'0000094F'
MQRC_AUTH_INFO_REC_ERROR	2384	X'00000950'
MQRC_AIR_ERROR	2385	X'00000951'
MQRC_AUTH_INFO_TYPE_ERROR	2386	X'00000952'
MQRC_AUTH_INFO_CONN_NAME_ERROR	2387	X'00000953'
MQRC_LDAP_USER_NAME_ERROR	2388	X'00000954'

Tabela 304. Valores de constantes (continuação)

Nome	Valor decimal	Valor Hexadecimal
MQRC_LDAP_USER_NAME_LENGTH_ERR	2389	X'00000955'
MQRC_LDAP_PASSWORD_ERROR	2390	X'00000956'
MQRC_SSL_ALREADY_INITIALIZED	2391	X'00000957'
MQRC_SSL_CONFIG_ERROR	2392	X'00000958'
MQRC_SSL_INITIALIZATION_ERROR	2393	X'00000959'
MQRC_Q_INDEX_TYPE_ERROR	2394	X'0000095A'
MQRC_CFBS_ERROR	2395	X'0000095B'
MQRC_SSL_NOT_ALLOWED	2396	X'0000095C'
MQRC_JSSE_ERROR	2397	X'0000095D'
MQRC_SSL_PEER_NAME_MISMATCH	2398	X'0000095E'
MQRC_SSL_PEER_NAME_ERROR	2399	X'0000095F'
MQRC_UNSUPPORTED_CIPHER_SUITE	2400	X'00000960'
MQRC_SSL_CERTIFICATE_REVOKED	2401	X'00000961'
MQRC_SSL_CERT_STORE_ERROR	2402	X'00000962'
MQRC_CLIENT_EXIT_LOAD_ERROR	2406	X'00000966'
MQRC_CLIENT_EXIT_ERROR	2407	X'00000967'
MQRC_UOW_COMMITTED	2408	X'00000968'
MQRC_SSL_KEY_RESET_ERROR	2409	X'00000969'
MQRC_UNKNOWN_COMPONENT_NAME	2410	X'0000096A'
MQRC_LOGGER_STATUS	2411	X'0000096B'
MQRC_COMMAND_MQSC	2412	X'0000096C'
MQRC_COMMAND_PCF	2413	X'0000096D'
MQRC_CFIF_ERROR	2414	X'0000096E'
MQRC_CFSF_ERROR	2415	X'0000096F'
MQRC_CFGR_ERROR	2416	X'00000970'
MQRC_MSG_NOT_ALLOWED_IN_GROUP	2417	X'00000971'
MQRC_FILTER_OPERATOR_ERROR	2418	X'00000972'
MQRC_NESTED_SELECTOR_ERROR	2419	X'00000973'
MQRC_EPH_ERROR	2420	X'00000974'
MQRC_RFH_FORMAT_ERROR	2421	X'00000975'
MQRC_CFBF_ERROR	2422	X'00000976'
MQRC_CLIENT_CHANNEL_CONFLICT	2423	X'00000977'
MQRC_SD_ERROR	2424	X'00000978'
MQRC_TOPIC_STRING_ERROR	2425	X'00000979'
MQRC_STS_ERROR	2426	X'0000097A'
MQRC_NO_SUBSCRIPTION	2428	X'0000097C'
MQRC_SUBSCRIPTION_IN_USE	2429	X'0000097D'
MQRC_STAT_TYPE_ERROR	2430	X'0000097E'
MQRC_SUB_USER_DATA_ERROR	2431	X'0000097F'

Tabela 304. Valores de constantes (continuação)

Nome	Valor decimal	Valor Hexadecimal
MQRC_SUB_ALREADY_EXISTS	2432	X'00000980'
MQRC_IDENTITY_MISMATCH	2434	X'00000982'
MQRC_ALTER_SUB_ERROR	2435	X'00000983'
MQRC_DURABILITY_NOT_ALLOWED	2436	X'00000984'
MQRC_NO_RETAINED_MSG	2437	X'00000985'
MQRC_SRO_ERROR	2438	X'00000986'
MQRC_SUB_NAME_ERROR	2440	X'00000988'
MQRC_OBJECT_STRING_ERROR	2441	X'00000989'
MQRC_PROPERTY_NAME_ERROR	2442	X'0000098A'
MQRC_SEGMENTATION_NOT_ALLOWED	2443	X'0000098B'
MQRC_CBD_ERROR	2444	X'0000098C'
MQRC_CTLO_ERROR	2445	X'0000098D'
MQRC_NO_CALLBACKS_ACTIVE	2446	X'0000098E'
MQRC_CALLBACK_NOT_REGISTERED	2448	X'00000990'
MQRC_OPTIONS_CHANGED	2457	X'00000999'
MQRC_READ_AHEAD_MSGS	2458	X'0000099A'
MQRC_SELECTOR_SYNTAX_ERROR	2459	X'0000099B'
MQRC_HMSG_ERROR	2460	X'0000099C'
MQRC_CMHO_ERROR	2461	X'0000099D'
MQRC_DMHO_ERROR	2462	X'0000099E'
MQRC_SMPO_ERROR	2463	X'0000099F'
MQRC_IMPO_ERROR	2464	X'000009A0'
MQRC_PROPERTY_NAME_TOO_BIG	2465	X'000009A1'
MQRC_PROP_VALUE_NOT_CONVERTED	2466	X'000009A2'
MQRC_PROP_TYPE_NOT_SUPPORTED	2467	X'000009A3'
MQRC_PROPERTY_VALUE_TOO_BIG	2469	X'000009A5'
MQRC_PROP_CONV_NOT_SUPPORTED	2470	X'000009A6'
MQRC_PROPERTY_NOT_AVAILABLE	2471	X'000009A7'
MQRC_PROP_NUMBER_FORMAT_ERROR	2472	X'000009A8'
MQRC_PROPERTY_TYPE_ERROR	2473	X'000009A9'
MQRC_PROPERTIES_TOO_BIG	2478	X'000009AE'
MQRC_PUT_NOT_RETIDO	2479	X'000009AF'
MQRC_ALIAS_TARGTYPE_CHANGED	2480	X'000009B0'
MQRC_DMPO_ERROR	2481	X'000009B1'
MQRC_PD_ERROR	2482	X'000009B2'
MQRC_CALLBACK_TYPE_ERROR	2483	X'000009B3'
MQRC_CBD_OPTIONS_ERROR	2484	X'000009B4'
MQRC_MAX_MSG_LENGTH_ERROR	2485	X'000009B5'
MQRC_CALLBACK_ROUTINE_ERROR	2486	X'000009B6'

Tabela 304. Valores de constantes (continuação)

Nome	Valor decimal	Valor Hexadecimal
MQRC_CALLBACK_LINK_ERROR	2487	X'000009B7'
MQRC_OPERATION_ERROR	2488	X'000009B8'
MQRC_BMHO_ERROR	2489	X'000009B9'
MQRC_UNSUPPORTED_PROPERTY	2490	X'000009BA'
MQRC_PROP_NAME_NOT_CONVERTED	2492	X'000009BC'
MQRC_GET_ENABLED	2494	X'000009BE'
MQRC_MODULE_NOT_FOUND	2495	X'000009BF'
MQRC_MODULE_INVALID	2496	X'000009C0'
MQRC_MODULE_ENTRY_NOT_FOUND	2497	X'000009C1'
MQRC_MIXED_CONTENT_NOT_ALLOWED	2498	X'000009C2'
MQRC_MSG_HANDLE_IN_USE	2499	X'000009C3'
MQRC_HCONN_ASYNC_ACTIVE	2500	X'000009C4'
MQRC_MHBO_ERROR	2501	X'000009C5'
MQRC_PUBLICATION_FAILURE	2502	X'000009C6'
MQRC_SUB_INHIBITED	2503	X'000009C7'
MQRC_SELECTOR_ALWAYS_FALSE	2504	X'000009C8'
MQRC_XEPO_ERROR	2507	X'000009CB'
MQRC_DURABILITY_NOT_ALTERABLE	2509	X'000009CD'
MQRC_TOPIC_NOT_ALTERABLE	2510	X'000009CE'
MQRC_SUBLEVEL_NOT_ALTERABLE	2512	X'000009D0'
MQRC_PROPERTY_NAME_LENGTH_ERR	2513	X'000009D1'
MQRC_DUPLICATE_GROUP_SUB	2514	X'000009D2'
MQRC_GROUPING_NOT_ALTERABLE	2515	X'000009D3'
MQRC_SELECTOR_INVALID_FOR_TYPE	2516	X'000009D4'
MQRC_HOBJ QUIESCED	2517	X'000009D5'
MQRC_HOBJ QUIESCED_NO_MSGS	2518	X'000009D6'
MQRC_SELECTION_STRING_ERROR	2519	X'000009D7'
MQRC_RES_OBJECT_STRING_ERROR	2520	X'000009D8'
MQRC_CONNECTION_SUSPENDED	2521	X'000009D9'
MQRC_INVALID_DESTINATION	2522	X'000009DA'
MQRC_INVALID_SUBSCRIPTION	2523	X'000009DB'
MQRC_SELECTOR_NOT_ALTERABLE	2524	X'000009DC'
MQRC_RETAINED_MSG_Q_ERROR	2525	X'000009DD'
MQRC_RETAINED_NOT_DELIVERED	2526	X'000009DE'
MQRC_RFH_RESTRICTED_FORMAT_ERR	2527	X'000009DF'
MQRC_CONNECTION_STOPPED	2528	X'000009E0'
MQRC_ASYNC_UOW_CONFLICT	2529	X'000009E1'
MQRC_ASYNC_XA_CONFLICT	2530	X'000009E2'
MQRC_PUBSUB_INHIBITED	2531	X'000009E3'

Tabela 304. Valores de constantes (continuação)

Nome	Valor decimal	Valor Hexadecimal
MQRC_MSG_HANDLE_COPY_FAILURE	2532	X'000009E4'
MQRC_DEST_CLASS_NOT_ALTERABLE	2533	X'000009E5'
MQRC_OPERATION_NOT_ALLOWED	2534	X'000009E6'
MQRC_ACTION_ERROR	2535	X'000009E7'
MQRC_CHANNEL_NOT_AVAILABLE	2537	X'000009E9'
MQRC_HOST_NOT_AVAILABLE	2538	X'000009EA'
MQRC_CHANNEL_CONFIG_ERROR	2539	X'000009EB'
MQRC_UNKNOWN_CHANNEL_NAME	2540	X'000009EC'
MQRC_LOOPING_PUBLICATION	2541	X'000009ED'
MQRC_ALREADY_JOINED	2542	X'000009EE'
MQRC_CHANNEL_SSL_WARNING	2552	X'000009F8'
MQRC_OCSP_URL_ERROR	2553	X'000009F9'
MQRC_CIPHER_SPEC_NOT_SUITE_B	2591	X'00000A1F'
MQRC_SUITE_B_ERROR	2592	X'00000A20'
MQRC_PASSWORD_PROTECTION_ERROR	2594	X'00000A22'
MQRC_REOPEN_EXCL_INPUT_ERROR	6100	X'000017D4'
MQRC_REOPEN_INQUIRE_ERROR	6101	X'000017D5'
MQRC_REOPEN_SAVED_CONTEXT_ERR	6102	X'000017D6'
MQRC_REOPEN_TEMPORARY_Q_ERROR	6103	X'000017D7'
MQRC_ATTRIBUTE_LOCKED	6104	X'000017D8'
MQRC_CURSOR_NOT_VALID	6105	X'000017D9'
MQRC_ENCODING_ERROR	6106	X'000017DA'
MQRC_STRUC_ID_ERROR	6107	X'000017DB'
MQRC_NULL_POINTER	6108	X'000017DC'
MQRC_NO_CONNECTION_REFERENCE	6109	X'000017DD'
MQRC_NO_BUFFER	6110	X'000017DE'
MQRC_BINARY_DATA_LENGTH_ERROR	6111	X'000017DF'
MQRC_BUFFER_NOT_AUTOMATIC	6112	X'000017E0'
MQRC_INSUFENT_BUFFER	6113	X'000017E1'
MQRC_INSUFENT_DATA	6114	X'000017E2'
MQRC_DATA_TRUNCATED	6115	X'000017E3'
MQRC_ZERO_LENGTH	6116	X'000017E4'
MQRC_NEGATIVE_LENGTH	6117	X'000017E5'
MQRC_NEGATIVE_OFFSET	6118	X'000017E6'
MQRC_INCONSISTENT_FORMAT	6119	X'000017E7'
MQRC_INCONSISTENT_OBJECT_STATE	6120	X'000017E8'
MQRC_CONTEXT_OBJECT_NOT_VALID	6121	X'000017E9'
MQRC_CONTEXT_OPEN_ERROR	6122	X'000017EA'
MQRC_STRUC_LENGTH_ERROR	6123	X'000017EB'

Tabela 304. Valores de constantes (continuação)

Nome	Valor decimal	Valor Hexadecimal
MQRC_NOT_CONNECTED	6124	X'000017EC'
MQRC_NOT_OPEN	6125	X'000017ED'
MQRC_DISTRIBUTION_LIST_EMPTY	6126	X'000017EE'
MQRC_INCONSISTENT_OPEN_OPTIONS	6127	X'000017EF'
MQRC_ERRONE_VERSION	6128	X'000017F0'
MQRC_REFERENCE_ERROR	6129	X'000017F1'

## MQRCCF\_\* (Códigos de razão do cabeçalho do formato de comando)

Para obter mais informações sobre a resposta do programador, consulte [Códigos de Razão PCF](#)

Tabela 305. Valores de constantes

Nome	Valor decimal	Valor Hexadecimal
MQRCCF_CFH_TYPE_ERROR	3001	X'00000BB9'
MQRCCF_CFH_LENGTH_ERROR	3002	X'00000BBA'
MQRCCF_CFH_VERSION_ERROR	3003	X'00000BBB'
MQRCCF_CFH_MSG_SEQ_NUMBER_ERR	3004	X'00000BBC'
MQRCCF_CFH_CONTROL_ERROR	3005	X'00000BBD'
MQRCCF_CFH_PARM_COUNT_ERROR	3006	X'00000BBE'
MQRCCF_CFH_COMMAND_ERROR	3007	X'00000BBF'
MQRCCF_COMMAND_FAILED	3008	X'00000BC0'
MQRCCF_CFIN_LENGTH_ERROR	3009	X'00000BC1'
MQRCCF_CFST_LENGTH_ERROR	3010	X'00000BC2'
MQRCCF_CFST_STRING_LENGTH_ERR	3011	X'00000BC3'
MQRCCF_FORCE_VALUE_ERROR	3012	X'00000BC4'
MQRCCF_STRUCTURE_TYPE_ERROR	3013	X'00000BC5'
MQRCCF_CFIN_PARM_ID_ERROR	3014	X'00000BC6'
MQRCCF_CFST_PARM_ID_ERROR	3015	X'00000BC7'
MQRCCF_MSG_LENGTH_ERROR	3016	X'00000BC8'
MQRCCF_CFIN_DUPLICATE_PARM	3017	X'00000BC9'
MQRCCF_CFST_DUPLICATE_PARM	3018	X'00000BCA'
MQRCCF_PARM_COUNT_TOO_SMALL	3019	X'00000BCB'
MQRCCF_PARM_COUNT_TOO_BIG	3020	X'00000BCC'
MQRCCF_Q_ALREADY_IN_CELL	3021	X'00000BCD'
MQRCCF_Q_TYPE_ERROR	3022	X'00000BCE'
MQRCCF_MD_FORMAT_ERROR	3023	X'00000BCF'
MQRCCF_CFSL_LENGTH_ERROR	3024	X'00000BD0'
MQRCCF_REPLACE_VALUE_ERROR	3025	X'00000BD1'
MQRCCF_CFIL_DUPLICATE_VALUE	3026	X'00000BD2'
MQRCCF_CFIL_COUNT_ERROR	3027	X'00000BD3'
MQRCCF_CFIL_LENGTH_ERROR	3028	X'00000BD4'



Tabela 305. Valores de constantes (continuação)

Nome	Valor decimal	Valor Hexadecimal
MQRCCF_QUIESCE_VALUE_ERROR	3029	X'00000BD5'
MQRCCF_MODE_VALUE_ERROR	3029	X'00000BD5'
MQRCCF_MSG_SEQ_NUMBER_ERROR	3030	X'00000BD6'
MQRCCF_PING_DATA_COUNT_ERROR	3031	X'00000BD7'
MQRCCF_PING_DATA_COMPARE_ERROR	3032	X'00000BD8'
MQRCCF_CFSL_PARM_ID_ERROR	3033	X'00000BD9'
MQRCCF_CHANNEL_TYPE_ERROR	3034	X'00000BDA'
MQRCCF_PARM_SEQUENCE_ERROR	3035	X'00000BDB'
MQRCCF_XMIT_PROTOCOL_TYPE_ERR	3036	X'00000BDC'
MQRCCF_BATCH_SIZE_ERROR	3037	X'00000BDD'
MQRCCF_DISC_INT_ERROR	3038	X'00000BDE'
MQRCCF_SHORT_RETRY_ERROR	3039	X'00000BDF'
MQRCCF_SHORT_TIMER_ERROR	3040	X'00000BE0'
MQRCCF_LONG_RETRY_ERROR	3041	X'00000BE1'
MQRCCF_LONG_TIMER_ERROR	3042	X'00000BE2'
MQRCCF_SEQ_NUMBER_WRAP_ERROR	3043	X'00000BE3'
MQRCCF_MAX_MSG_LENGTH_ERROR	3044	X'00000BE4'
MQRCCF_PUT_AUTH_ERROR	3045	X'00000BE5'
MQRCCF_PURGE_VALUE_ERROR	3046	X'00000BE6'
MQRCCF_CFIL_PARM_ID_ERROR	3047	X'00000BE7'
MQRCCF_MSG_TRUNCATED	3048	X'00000BE8'
MQRCCF_CCSID_ERROR	3049	X'00000BE9'
MQRCCF_ENCODING_ERROR	3050	X'00000BEA'
MQRCCF_QUEUES_VALUE_ERROR	3051	X'00000BEB'
MQRCCF_DATA_CONV_VALUE_ERROR	3052	X'00000BEC'
MQRCCF_INDOUBT_VALUE_ERROR	3053	X'00000BED'
MQRCCF_ESCAPE_TYPE_ERROR	3054	X'00000BEE'
MQRCCF_REPOS_VALUE_ERROR	3055	X'00000BEF'
MQRCCF_CHANNEL_TABLE_ERROR	3062	X'00000BF6'
MQRCCF_MCA_TYPE_ERROR	3063	X'00000BF7'
MQRCCF_CHL_INST_TYPE_ERROR	3064	X'00000BF8'
MQRCCF_CHL_STATUS_NOT_FOUND	3065	X'00000BF9'
MQRCCF_CFSL_DUPLICATE_PARM	3066	X'00000BFA'
MQRCCF_CFSL_TOTAL_LENGTH_ERROR	3067	X'00000BFB'
MQRCCF_CFSL_COUNT_ERROR	3068	X'00000BFC'
MQRCCF_CFSL_STRING_LENGTH_ERR	3069	X'00000BFD'
MQRCCF_BROKER_DELETED	3070	X'00000BFE'
MQRCCF_STREAM_ERROR	3071	X'00000BFF'
MQRCCF_TOPIC_ERROR	3072	X'00000C00'

Tabela 305. Valores de constantes (continuação)

Nome	Valor decimal	Valor Hexadecimal
MQRCCF_NOT_REGISTERED	3073	X'00000C01'
MQRCCF_Q_MGR_NAME_ERROR	3074	X'00000C02'
MQRCCF_INCORRECT_STREAM	3075	X'00000C03'
MQRCCF_Q_NAME_ERROR	3076	X'00000C04'
MQRCCF_NO_RETAINED_MSG	3077	X'00000C05'
MQRCCF_DUPLICATE_IDENTITY	3078	X'00000C06'
MQRCCF_INCORRECT_Q	3079	X'00000C07'
MQRCCF_CORREL_ID_ERROR	3080	X'00000C08'
MQRCCF_NOT_AUTHORIZED	3081	X'00000C09'
MQRCCF_UNKNOWN_STREAM	3082	X'00000C0A'
MQRCCF_REG_OPTIONS_ERROR	3083	X'00000C0B'
MQRCCF_PUB_OPTIONS_ERROR	3084	X'00000C0C'
MQRCCF_UNKNOWN_BROKER	3085	X'00000C0D'
MQRCCF_Q_MGR_CCSID_ERROR	3086	X'00000C0E'
MQRCCF_DEL_OPTIONS_ERROR	3087	X'00000C0F'
MQRCCF_CLUSTER_NAME_CONFLICT	3088	X'00000C10'
MQRCCF_REPOS_NAME_CONFLICT	3089	X'00000C11'
MQRCCF_CLUSTER_Q_USAGE_ERROR	3090	X'00000C12'
MQRCCF_ACTION_VALUE_ERROR	3091	X'00000C13'
MQRCCF_COMMS_LIBRARY_ERROR	3092	X'00000C14'
MQRCCF_NETBIOS_NAME_ERROR	3093	X'00000C15'
MQRCCF_BROKER_COMMAND_FAILED	3094	X'00000C16'
MQRCCF_CFST_CONFLICTING_PARM	3095	X'00000C17'
MQRCCF_PATH_NOT_VALID	3096	X'00000C18'
MQRCCF_PARM_SYNTAX_ERROR	3097	X'00000C19'
MQRCCF_PWD_LENGTH_ERROR	3098	X'00000C1A'
MQRCCF_FILTER_ERROR	3150	X'00000C4E'
MQRCCF_EQUIVOC_USER	3151	X'00000C4F'
MQRCCF_DUPLICATE_SUBSCRIPTION	3152	X'00000C50'
MQRCCF_SUB_NAME_ERROR	3153	X'00000C51'
MQRCCF_SUB_IDENTITY_ERROR	3154	X'00000C52'
MQRCCF_SUBSCRIPTION_IN_USE	3155	X'00000C53'
MQRCCF_SUBSCRIPTION_LOCKED	3156	X'00000C54'
MQRCCF_ALREADY_ENTROU	3157	X'00000C55'
MQRCCF_OBJECT_IN_USE	3160	X'00000C58'
MQRCCF_UNKNOWN_FILE_NAME	3161	X'00000C59'
MQRCCF_FILE_NOT_AVAILABLE	3162	X'00000C5A'
MQRCCF_DISC_RETRY_ERROR	3163	X'00000C5B'
MQRCCF_ALLOC_RETRY_ERROR	3164	X'00000C5C'

Tabela 305. Valores de constantes (continuação)

Nome	Valor decimal	Valor Hexadecimal
MQRCCF_ALLOC_SLOW_TIMER_ERROR	3165	X'00000C5D'
MQRCCF_ALLOC_FAST_TIMER_ERROR	3166	X'00000C5E'
MQRCCF_PORT_NUMBER_ERROR	3167	X'00000C5F'
MQRCCF_CHL_SYSTEM_NOT_ACTIVE	3168	X'00000C60'
MQRCCF_ENTITY_NAME_MISSING	3169	X'00000C61'
MQRCCF_PROFILE_NAME_ERROR	3170	X'00000C62'
MQRCCF_AUTH_VALUE_ERROR	3171	X'00000C63'
MQRCCF_AUTH_VALUE_MISSING	3172	X'00000C64'
MQRCCF_OBJECT_TYPE_MISSING	3173	X'00000C65'
MQRCCF_CONNECTION_ID_ERROR	3174	X'00000C66'
MQRCCF_LOG_TYPE_ERROR	3175	X'00000C67'
MQRCCF_PROGRAM_NOT_AVAILABLE	3176	X'00000C68'
MQRCCF_PROGRAM_AUTH_FAILED	3177	X'00000C69'
MQRCCF_NONE_FOUND	3200	X'00000C80'
MQRCCF_SECURITY_SWITCH_OFF	3201	X'00000C81'
MQRCCF_SECURITY_REFRESH_FAILED	3202	X'00000C82'
MQRCCF_PARM_CONFLICT	3203	X'00000C83'
MQRCCF_COMMAND_INIBITED	3204	X'00000C84'
MQRCCF_OBJECT_BEING_DELETED	3205	X'00000C85'
MQRCCF_STORAGE_CLASS_IN_USE	3207	X'00000C87'
MQRCCF_OBJECT_NAME_RESTRICTED	3208	X'00000C88'
MQRCCF_OBJECT_LIMIT_EXCEDENTE	3209	X'00000C89'
MQRCCF_OBJECT_OPEN_FORCE	3210	X'00000C8A'
MQRCCF_DISPOSITION_CONFLICT	3211	X'00000C8B'
MQRCCF_Q_MGR_NOT_IN_QSG	3212	X'00000C8C'
MQRCCF_ATTR_VALUE_FIXED	3213	X'00000C8D'
MQRCCF_NAMELIST_ERROR	3215	X'00000C8F'
MQRCCF_NO_CHANNEL_INITIATOR	3217	X'00000C91'
MQRCCF_CHANNEL_INITIATOR_ERROR	3218	X'00000C92'
MQRCCF_COMMAND_LEVEL_CONFLICT	3222	X'00000C96'
MQRCCF_Q_ATTR_CONFLICT	3223	X'00000C97'
MQRCCF_EVENTS_DISABLED	3224	X'00000C98'
MQRCCF_COMMAND_SCOPE_ERROR	3225	X'00000C99'
MQRCCF_COMMAND_REPLY_ERROR	3226	X'00000C9A'
MQRCCF_FUNCTION_RESTRICTED	3227	X'00000C9B'
MQRCCF_PARM_MISSING	3228	X'00000C9C'
MQRCCF_PARM_VALUE_ERROR	3229	X'00000C9D'
MQRCCF_COMMAND_LENGTH_ERROR	3230	X'00000C9E'
MQRCCF_COMMAND_ORIGIN_ERROR	3231	X'00000C9F'

Tabela 305. Valores de constantes (continuação)

Nome	Valor decimal	Valor Hexadecimal
MQRCCF_LISTENER_CONFLICT	3232	X'00000CA0'
MQRCCF_LISTENER_STARTED	3233	X'00000CA1'
MQRCCF_LISTENER_STOPPED	3234	X'00000CA2'
MQRCCF_CHANNEL_ERROR	3235	X'00000CA3'
MQRCCF_CF_STRUC_ERROR	3236	X'00000CA4'
MQRCCF_UNKNOWN_USER_ID	3237	X'00000CA5'
MQRCCF_UNEXPECTED_ERROR	3238	X'00000CA6'
MQRCCF_NO_XCF_PARTNER	3239	X'00000CA7'
MQRCCF_CFGR_PARM_ID_ERROR	3240	X'00000CA8'
MQRCCF_CFIF_LENGTH_ERROR	3241	X'00000CA9'
MQRCCF_CFIF_OPERATOR_ERROR	3242	X'00000CAA'
MQRCCF_CFIF_PARM_ID_ERROR	3243	X'00000CAB'
MQRCCF_CFSF_FILTER_VAL_LEN_ERR	3244	X'00000CAC'
MQRCCF_CFSF_LENGTH_ERROR	3245	X'00000CAD'
MQRCCF_CFSF_OPERATOR_ERROR	3246	X'00000CAE'
MQRCCF_CFSF_PARM_ID_ERROR	3247	X'00000CAF'
MQRCCF_TOO_MANY_FILTERS	3248	X'00000CB0'
MQRCCF_LISTENER_RUNNING	3249	X'00000CB1'
MQRCCF_LSTR_STATUS_NOT_FOUND	3250	X'00000CB2'
MQRCCF_SERVICE_RUNNING	3251	X'00000CB3'
MQRCCF_SERV_STATUS_NOT_FOUND	3252	X'00000CB4'
MQRCCF_SERVICE_STOPPED	3253	X'00000CB5'
MQRCCF_CFBS_DUPLICATE_PARM	3254	X'00000CB6'
MQRCCF_CFBS_LENGTH_ERROR	3255	X'00000CB7'
MQRCCF_CFBS_PARM_ID_ERROR	3256	X'00000CB8'
MQRCCF_CFBS_STRING_LENGTH_ERR	3257	X'00000CB9'
MQRCCF_CFGR_LENGTH_ERROR	3258	X'00000CBA'
MQRCCF_CFGR_PARM_COUNT_ERROR	3259	X'00000CBB'
MQRCCF_CONN_NOT_STOPPED	3260	X'00000CBC'
MQRCCF_SERVICE_REQUEST_PENDING	3261	X'00000CBD'
MQRCCF_NO_START_CMD	3262	X'00000CBE'
MQRCCF_NO_STOP_CMD	3263	X'00000CBF'
MQRCCF_CFBF_LENGTH_ERROR	3264	X'00000CC0'
MQRCCF_CFBF_PARM_ID_ERROR	3265	X'00000CC1'
MQRCCF_CFBF_OPERATOR_ERROR	3266	X'00000CC2'
MQRCCF_CFBF_FILTER_VAL_LEN_ERR	3267	X'00000CC3'
MQRCCF_LISTENER_STILL_ACTIVE	3268	X'00000CC4'
MQRCCF_DEF_XMIT_Q_CLUS_ERROR	3269	X'00000CC5'
MQRCCF_TOPICSTR_ALREADY_EXISTS	3300	X'00000CE4'

Tabela 305. Valores de constantes (continuação)


Nome	Valor decimal	Valor Hexadecimal
MQRCCF_SHARING_CONVS_ERROR	3301	X'00000CE5'
MQRCCF_SHARING_CONVS_TYPE	3302	X'00000CE6'
MQRCCF_SECURITY_CASE_CONFLICT	3303	X'00000CE7'
MQRCCF_TOPIC_TYPE_ERROR	3305	X'00000CE9'
MQRCCF_MAX_INSTANCES_ERROR	3306	X'00000CEA'
MQRCCF_MAX_INSTS_PER_CLNT_ERR	3307	X'00000CEB'
MQRCCF_TOPIC_STRING_NOT_FOUND	3308	X'00000CEC'
MQRCCF_SUBSCRIPTION_POINT_ERR	3309	X'00000CED'
MQRCCF_SUB_ALREADY_EXISTS	3311	X'00000CEF'
MQRCCF_UNKNOWN_OBJECT_NAME	3312	X'00000CF0'
MQRCCF_REMOTE_Q_NAME_ERROR	3313	X'00000CF1'
MQRCCF_DURABILITY_NOT_ALLOWED	3314	X'00000CF2'
MQRCCF_HOBJ_ERROR	3315	X'00000CF3'
MQRCCF_DEST_NAME_ERROR	3316	X'00000CF4'
MQRCCF_INVALID_DESTINATION	3317	X'00000CF5'
MQRCCF_PUBSUB_INIBIDA	3318	X'00000CF6'
MQRCCF_CHLAUTH_TYPE_ERROR	3326	X'00000CFE'
MQRCCF_CHLAUTH_ACTION_ERROR	3327	X'00000CFF'
MQRCCF_CHLAUTH_USERSRC_ERROR	3335	X'00000D07'
MQRCCF_ERRR_CHLAUTH_TYPE	3336	X'00000D08'
MQRCCF_CHLAUTH_ALREADY_EXISTS	3337	X'00000D09'
MQRCCF_CHLAUTH_NOT_FOUND	3338	X'00000D0A'
MQRCCF_ERRRADA_chlauth_action	3339	X'00000D0B'
MQRCCF_ERRR_CHLAUTH_USERSRC	3340	X'00000D0C'
MQRCCF_CHLAUTH_WARN_ERROR	3341	X'00000D0D'
MQRCCF_ERRR_CHLAUTH_MATCH	3342	X'00000D0E'
MQRCCF_IPADDR_RANGE_CONFLICT	3343	X'00000D0F'
MQRCCF_CHLAUTH_MAX_EXCEDIDO	3344	X'00000D10'
MQRCCF_IPADDR_ERROR	3345	X'00000D11'
MQRCCF_IPADDR_RANGE_ERROR	3346	X'00000D12'
MQRCCF_PROFILE_NAME_MISSING	3347	X'00000D13'
MQRCCF_CHLAUTH_CLNTUSER_ERROR	3348	X'00000D14'
MQRCCF_CHLAUTH_NAME_ERROR	3349	X'00000D15'
MQRCCF_SUITE_B_ERROR	3353	X'00000D19'
MQRCCF_PSCLUS_DISABLED_TOPDEF	3359	X'00000D1F'
MQRCCF_PSCLUS_TOPIC_EXISTS	3360	X'00000D20'
MQRCCF_INVALID_PROTOCOL	3365	X'00000D25'
MQRCCF_ACCESS_BLOCKED	3382	X'00000D36'
MQRCCF_OBJECT_ALREADY_EXISTS	4001	X'00000FA1'

Tabela 305. Valores de constantes (continuação)

Nome	Valor decimal	Valor Hexadecimal
MQRCCF_OBJECT_ERR_TYPE	4002	X'0000FA2'
MQRCCF_LIKE_OBJECT_ERR_TYPE	4003	X'0000FA3'
MQRCCF_OBJECT_OPEN	4004	X'0000FA4'
MQRCCF_ATTR_VALUE_ERROR	4005	X'0000FA5'
MQRCCF_UNKNOWN_Q_MGR	4006	X'0000FA6'
MQRCCF_Q_ERRONE_TYPE	4007	X'0000FA7'
MQRCCF_OBJECT_NAME_ERROR	4008	X'0000FA8'
MQRCCF_ALLOCATE_FAILED	4009	X'0000FA9'
MQRCCF_HOST_NOT_AVAILABLE	4010	X'0000FAA'
MQRCCF_CONFIGURATION_ERROR	4011	X'0000FAB'
MQRCCF_CONNECTION_RECUSADO	4012	X'0000FAC'
MQRCCF_ENTRY_ERROR	4013	X'0000FAD'
MQRCCF_SEND_FAILED	4014	X'0000FAE'
MQRCCF_RECEIVED_DATA_ERROR	4015	X'0000FAF'
MQRCCF_RECEIVE_FAILED	4016	X'0000FB0'
MQRCCF_CONNECTION_CLOSED	4017	X'0000FB1'
MQRCCF_NO_STORAGE	4018	X'0000FB2'
MQRCCF_NO_COMMS_MANAGER	4019	X'0000FB3'
MQRCCF_LISTENER_NOT_STARTED	4020	X'0000FB4'
MQRCCF_BIND_FAILED	4024	X'0000FB8'
MQRCCF_CHANNEL_INDOUBT	4025	X'0000FB9'
MQRCCF_MQCONN_FAILED	4026	X'0000FBA'
MQRCCF_MQOPEN_FAILED	4027	X'0000FBB'
MQRCCF_MQGET_FAILED	4028	X'0000FBC'
MQRCCF_MQPUT_FAILED	4029	X'0000FBD'
MQRCCF_PING_ERROR	4030	X'0000FBE'
MQRCCF_CHANNEL_IN_USE	4031	X'0000FBF'
MQRCCF_CHANNEL_NOT_FOUND	4032	X'0000FC0'
MQRCCF_UNKNOWN_REMOTE_CHANNEL	4033	X'0000FC1'
MQRCCF_REMOTE_QM_UNAVAILABLE	4034	X'0000FC2'
MQRCCF_REMOTE_QM_TERMINATING	4035	X'0000FC3'
MQRCCF_MQINQ_FAILED	4036	X'0000FC4'
MQRCCF_NOT_XMIT_Q	4037	X'0000FC5'
MQRCCF_CHANNEL_DISABLED	4038	X'0000FC6'
MQRCCF_USER_EXIT_NOT_AVAILABLE	4039	X'0000FC7'
MQRCCF_COMMIT_FAILED	4040	X'0000FC8'
MQRCCF_ERRONE_CHANNEL_TYPE	4041	X'0000FC9'
MQRCCF_CHANNEL_ALREADY_EXISTS	4042	X'0000FCA'
MQRCCF_DATA_TOO_LARGE	4043	X'0000FCB'

Tabela 305. Valores de constantes (continuação)

Nome	Valor decimal	Valor Hexadecimal
MQRCCF_CHANNEL_NAME_ERROR	4044	X'0000FCC'
MQRCCF_XMIT_Q_NAME_ERROR	4045	X'0000FCD'
MQRCCF_MCA_NAME_ERROR	4047	X'0000FCF'
MQRCCF_SEND_EXIT_NAME_ERROR	4048	X'0000FD0'
MQRCCF_SEC_EXIT_NAME_ERROR	4049	X'0000FD1'
MQRCCF_MSG_EXIT_NAME_ERROR	4050	X'0000FD2'
MQRCCF_RCV_EXIT_NAME_ERROR	4051	X'0000FD3'
MQRCCF_XMIT_Q_NAME_ERR_TYPE	4052	X'0000FD4'
MQRCCF_MCA_NAME_ERRONE_TYPE	4053	X'0000FD5'
MQRCCF_DISC_INT_ERRAD_TYPE	4054	X'0000FD6'
MQRCCF_SHORT_RETRY_ERRONE_TYPE	4055	X'0000FD7'
MQRCCF_SHORT_TIMER_ERRONE_TYPE	4056	X'0000FD8'
MQRCCF_LONG_RETRY_ERRONE_TYPE	4057	X'0000FD9'
MQRCCF_LONG_TIMER_ERRONE_TYPE	4058	X'0000FDA'
MQRCCF_PUT_AUTH_ERRONE_TYPE	4059	X'0000FDB'
MQRCCF_KEEP_ALIVE_INT_ERROR	4060	X'0000FDC'
MQRCCF_MISSING_CONN_NAME	4061	X'0000FDD'
MQRCCF_CONN_NAME_ERROR	4062	X'0000FDE'
MQRCCF_MQSET_FAILED	4063	X'0000FDF'
MQRCCF_CHANNEL_NOT_ACTIVE	4064	X'0000FE0'
MQRCCF_TERMINATED_BY_SEC_EXIT	4065	X'0000FE1'
MQRCCF_DYNAMIC_Q_SCOPE_ERROR	4067	X'0000FE3'
MQRCCF_CELL_DIR_NOT_AVAILABLE	4068	X'0000FE4'
MQRCCF_MR_COUNT_ERROR	4069	X'0000FE5'
MQRCCF_MR_COUNT_ERR_TYPE	4070	X'0000FE6'
MQRCCF_MR_EXIT_NAME_ERROR	4071	X'0000FE7'
MQRCCF_MR_EXIT_NAME_ERRONE_TYPE	4072	X'0000FE8'
MQRCCF_MR_INTERVAL_ERROR	4073	X'0000FE9'
MQRCCF_MR_INTERVAL_ERRONE_TYPE	4074	X'0000FEA'
MQRCCF_NPM_SPEED_ERROR	4075	X'0000FEB'
MQRCCF_NPM_SPEED_ERRONE_TYPE	4076	X'0000FEC'
MQRCCF_HB_INTERVAL_ERROR	4077	X'0000FED'
MQRCCF_HB_INTERVAL_ERRONE_TYPE	4078	X'0000FEE'
MQRCCF_CHAD_ERROR	4079	X'0000FEF'
MQRCCF_CHAD_ERRONE_TYPE	4080	X'0000FF0'
MQRCCF_CHAD_EVENT_ERROR	4081	X'0000FF1'
MQRCCF_CHAD_EVENT_ERRONE_TYPE	4082	X'0000FF2'
MQRCCF_CHAD_EXIT_ERROR	4083	X'0000FF3'
MQRCCF_CHAD_EXIT_ERRAD_TYPE	4084	X'0000FF4'

Tabela 305. Valores de constantes (continuação)		
Nome	Valor decimal	Valor Hexadecimal
MQRCCF_SUPPRESSED_BY_EXIT	4085	X'00000FF5'
MQRCCF_BATCH_INT_ERROR	4086	X'00000FF6'
MQRCCF_BATCH_INT_ERRR_TYPE	4087	X'00000FF7'
MQRCCF_NET_PRIORITY_ERROR	4088	X'00000FF8'
MQRCCF_NET_PRIORITY_ERRONE_TYPE	4089	X'00000FF9'
MQRCCF_CHANNEL_CLOSED	4090	X'00000FFA'
MQRCCF_Q_STATUS_NOT_FOUND	4091	X'00000FFB'
MQRCCF_SSL_CIPHER_SPEC_ERROR	4092	X'00000FFC'
MQRCCF_SSL_PEER_NAME_ERROR	4093	X'00000FFD'
MQRCCF_SSL_CLIENT_AUTH_ERROR	4094	X'00000FFE'
MQRCCF_RETAINED_NOT_SUPPORTED	4095	X'00000FFF'
 MQRCCF_KWD_VALUE_ERR_TYPE	4096	X'00001000'

### MQRCN\_\* (Constantes de reconexão do cliente).

Tabela 306. Valores de constantes		
Nome	Valor decimal	Valor Hexadecimal
MQRCN_NO	0	X'00000000'
MQRCN_YES	1	X'00000001'
MQRCN_Q_MGR	2	X'00000002'
MQRCN_DISABLED	3	X'00000003'

### MQRCVTIME\_\* (receber tipos de tempo limite)

Tabela 307. Valores de constantes		
Nome	Valor decimal	Valor Hexadecimal
MQRCVTIME_MULTIPLY	0	X'00000000'
MQRCVTIME_ADD	1	X'00000001'
MQRCVTIME_EQUAL	2	X'00000002'

### MQREADA\_\* (Valores de Leitura Antecipada)

Tabela 308. Valores de constantes		
Nome	Valor decimal	Valor Hexadecimal
MQREADA_NO	0	X'00000000'
MQREADA_YES	1	X'00000001'
MQREADA_DISABLED	2	X'00000002'
MQREADA_INIBIDO	3	X'00000003'
MQREADA_BACKLOG	4	X'00000004'



## MQRECORDING\_\* (Opções de gravação)

Tabela 309. Valores de constantes		
Nome	Valor decimal	Valor Hexadecimal
MQRECORDING_DISABLED	0	X'00000000'
MQRECORDING_Q	1	X'00000001'
MQRECORDING_MSG	2	X'00000002'

## MQREGO\_\* (Opções de Registro de Publicação / Assinatura)

Tabela 310. Valores de constantes		
Nome	Valor decimal	Valor Hexadecimal
MQREGO_NONE	0	X'00000000'
MQREGO_CORREL_ID_AS_IDENTITY	1	X'00000001'
MQREGO_ANONYMOUS	2	X'00000002'
MQREGO_LOCAL	4	X'00000004'
MQREGO_DIRECT_REQUESTS	8	X'00000008'
MQREGO_NEW_PUBLICATIONS_ONLY	16	X'00000010'
MQREGO_PUBLISH_ON_REQUEST_ONLY	32	X'00000020'
MQREGO_DEREGISTER_ALL	64	X'00000040'
MQREGO_INCLUDE_STREAM_NAME	128	X'00000080'
MQREGO_INFORM_IF_RETIDO	256	X'00000100'
MQREGO_DUPLICATES_OK	512	X'00000200'
MQREGO_NON_PERSISTENT	1024	X'00000400'
MQREGO_PERSISTENT	2048	X'00000800'
MQREGO_PERSISTENT_AS_PUBLISH	4096	X'00001000'
MQREGO_PERSISTENT_AS_Q	8192	X'00002000'
MQREGO_ADD_NAME	16384	X'00004000'
MQREGO_NO_ALTERATION	32768	X'00008000'
MQREGO_FULL_RESPONSE	65536	X'00010000'
MQREGO_JOIN_SHARED	131072	X'00020000'
MQREGO_JOIN_EXCLUSIVE	262144	X'00040000'
MQREGO_LEAVE_ONLY	524288	X'00080000'
MQREGO_VARIABLE_USER_ID	1048576	X'00100000'
MQREGO_LOCKED	2097152	X'00200000'

## MQRFH\_\* (Estrutura do cabeçalho de regras e formatação e Sinalizadores)

### Estrutura do cabeçalho de regras e formatação

Tabela 311. Estruturas de constantes	
Nome	Estrutura
MQRFH_STRUC_ID	"RFH~"
MQRFH_STRUC_ID_ARRAY	'R','F','H','~'

**Nota:** O símbolo – representa um único caractere em branco.

*Tabela 312. Valores de constantes*

Nome	Valor decimal	Valor Hexadecimal
MQRFH_VERSION_1	1	X'00000001'
MQRFH_VERSION_2	2	X'00000002'
MQRFH_STRUC_LENGTH_FIXED	32	X'00000020'
MQRFH_STRUC_LENGTH_FIXED_2	36	X'00000024'

## Regras e Cabeçalho de Formatação

*Tabela 313. Valores de constantes*

Nome	Valor decimal	Valor Hexadecimal
MQRFH_NONE	0	X'00000000'
MQRFH_NO_FLAGS	0	X'00000000'

## MQRFH2\_\* (Tag de opções de publicação / assinatura RFH2 Tags da pasta de nível superior)

*Tabela 314. Valores de constantes*

Nome	Valor decimal	Valor Hexadecimal
MQRFH2_NAME_VALUE_VERSION	1	X'00000001'

## MQRFH2\_\* (Nomes de Tag de Opções de Publicação / Assinatura)

MQRFH2_PUBSUB_CMD_FOLDER	"psc"
MQRFH2_PUBSUB_RESP_FOLDER	"pscr"
MQRFH2_MSG_CONTENT_FOLDER	"mcd"
MQRFH2_USER_FOLDER	"usr"

## MQRFH2\_\* (Nomes de tags XML de tag de publicação / assinatura)

MQRFH2_PUBSUB_CMD_FOLDER_B	"<psc>"
MQRFH2_PUBSUB_CMD_FOLDER_E	"</psc>"
MQRFH2_PUBSUB_RESP_FOLDER_B	"<pscr>"
MQRFH2_PUBSUB_RESP_FOLDER_E	"</pscr>"
MQRFH2_MSG_CONTENT_FOLDER_B	"<mcd>"
MQRFH2_MSG_CONTENT_FOLDER_E	"</mcd>"
MQRFH2_USER_FOLDER_B	"<usr>"
MQRFH2_USER_FOLDER_E	"</usr>"

## MQRL\_\* (Comprimento retornado)

*Tabela 315. Valores de constantes*

Nome	Valor decimal	Valor Hexadecimal
MQRL_UNDEFINED	-1	X'FFFFFFFF'

## MQRMH\_\* (Estrutura do cabeçalho da mensagem de referência)

*Tabela 316. Estruturas de constantes*

Nome	Estrutura
MQRMH_STRUC_ID	"RMH↵"
MQRMH_STRUC_ID_ARRAY	'R', 'M', 'H', '↵'

**Nota:** O símbolo ↵ representa um único caractere em branco.

*Tabela 317. Valores de constantes*

Nome	Valor decimal	Valor Hexadecimal
MQRMH_VERSION_1	1	X'00000001'
MQRMH_CURRENT_VERSION	1	X'00000001'

## MQRMHF\_\* (Sinalizadores do cabeçalho da mensagem de referência)

*Tabela 318. Valores de constantes*

Nome	Valor decimal	Valor Hexadecimal
MQRMHF_LAST	1	X'00000001'
MQRMHF_NOT_LAST	0	X'00000000'

## MQRO\_\* (Opções de Relatório).

*Tabela 319. Valores de constantes*

Nome	Valor decimal	Valor Hexadecimal
MQRO_EXCEPTION	16777216	X'01000000'
MQRO_EXCEPTION_WITH_DATA	50331648	X'03000000'
MQRO_EXCEPTION_WITH_FULL_DATA	117440512	X'07000000'
MQRO_EXPIRATION	2097152	X'00200000'
MQRO_EXPIRATION_WITH_DATA	6291456	X'00600000'
MQRO_EXPIRATION_WITH_FULL_DATA	14680064	X'00E00000'
MQRO_COA	256	X'00000100'
MQRO_COA_WITH_DATA	768	X'00000300'
MQRO_COA_WITH_FULL_DATA	1792	X'00000700'
MQRO_COD	2048	X'00000800'
MQRO_COD_WITH_DATA	6144	X'00001800'
MQRO_COD_WITH_FULL_DATA	14336	X'00003800'
MQRO_PAN	1	X'00000001'
MQRO_NAN	2	X'00000002'
MQRO_ACTIVITY	4	X'00000004'
MQRO_NEW_MSG_ID	0	X'00000000'
MQRO_PASS_MSG_ID	128	X'00000080'
MQRO_COPY_MSG_ID_TO_CORREL_ID	0	X'00000000'
MQRO_PASS_CORREL_ID	64	X'00000040'
MQRO_DEAD_LETTER_Q	0	X'00000000'
MQRO_DISCARD_MSG	134217728	X'08000000'

<i>Tabela 319. Valores de constantes (continuação)</i>		
Nome	Valor decimal	Valor Hexadecimal
MQRO_PASS_DISCARD_AND_EXPIRY	16384	X'00004000'
MQRO_NONE	0	X'00000000'

### **MQRO\_\* (Máscaras de Opções de Relatório)**

<i>Tabela 320. Valores de constantes</i>		
Nome	Valor decimal	Valor Hexadecimal
MQRO_REJECT_UNSUP_MASK	270270464	X'101C0000'
MQRO_ACCEPT_UNSUP_MASK	-270532353	X'EFE000FF'
MQRO_ACCEPT_UNSUP_IF_XMIT_MASK	261888	X'0003FF00'

### **MQROUTE\_\* (Trace-route)**

#### **Máximo de Atividades de Rastreamento de Rota (MQIACF\_MAX\_ACTIVIDADES)**

<i>Tabela 321. Valores de constantes</i>		
Nome	Valor decimal	Valor Hexadecimal
MQROUTE_UNLIMITED_ACTIVITIES	0	X'00000000'

#### **Detalhe de rastreamento de rotas (MQIACF\_ROUTE\_DETAIL)**

<i>Tabela 322. Valores de constantes</i>		
Nome	Valor decimal	Valor Hexadecimal
MQROUTE_DETAIL_LOW	2	X'00000002'
MQROUTE_DETAIL_MEDIUM	8	X'00000008'
MQROUTE_DETAIL_HIGH	32	X'00000020'

#### **Encaminhamento de rastreamento de rotas (MQIACF\_ROUTE\_FORWARDING)**

<i>Tabela 323. Valores de constantes</i>		
Nome	Valor decimal	Valor Hexadecimal
MQROUTE_FORWARD_ALL	256	X'00000100'
MQROUTE_FORWARD_IF_SUPPORTED	512	X'00000200'
MQROUTE_FORWARD_REJ_UNSUP_MASK	-65536	X'FFFF0000'

#### **Entrega de rastreamento de rotas (MQIACF\_ROUTE\_DELIVERY)**

<i>Tabela 324. Valores de constantes</i>		
Nome	Valor decimal	Valor Hexadecimal
MQROUTE_DELIVER_YES	4096	X'00001000'
MQROUTE_DELIVER_NO	8192	X'00002000'
MQROUTE_DELIVER_REJ_UNSUP_MASK	-65536	X'FFFF0000'

## Acumulação de rastreo de rotas (MQIACF\_ROUTE\_ACUMULAÇÃO)

*Tabela 325. Valores de constantes*

Nome	Valor decimal	Valor Hexadecimal
MQROUTE_ACCUMULATE_NONE	65539	X'00010003'
MQROUTE_ACCUMULATE_IN_MSG	65540	X'00010004'
MQROUTE_ACCUMULATE_AND_REPLY	65541	X'00010005'

## MQRP\_\* (Opções de Substituição de Formato de Comandos)..

*Tabela 326. Valores de constantes*

Nome	Valor decimal	Valor Hexadecimal
MQRP_YES	1	X'00000001'
MQRP_NO	0	X'00000000'

## MQRQ\_\* (Qualificadores de Motivo do formato de comando).

*Tabela 327. Valores de constantes*

Nome	Valor decimal	Valor Hexadecimal
MQRQ_CONN_NOT_AUTHORIZED	1	X'00000001'
MQRQ_OPEN_NOT_AUTHORIZED	2	X'00000002'
MQRQ_CLOSE_NOT_AUTHORIZED	3	X'00000003'
MQRQ_CMD_NOT_AUTHORIZED	4	X'00000004'
MQRQ_Q_MGR_STOPPING	5	X'00000005'
MQRQ_Q_MGR QUIESCING	6	X'00000006'
MQRQ_CHANNEL_STOPPED_OK	7	X'00000007'
MQRQ_CHANNEL_STOPPED_ERROR	8	X'00000008'
MQRQ_CHANNEL_STOPPED_RETRY	9	X'00000009'
MQRQ_CHANNEL_STOPPED_DISABLED	10	X'0000000A'
MQRQ_BRIDGE_STOPPED_OK	11	X'0000000B'
MQRQ_BRIDGE_STOPPED_ERROR	12	X'0000000C'
MQRQ_SSL_HANDSHAKE_ERROR	13	X'0000000D'
MQRQ_SSL_CIPHER_SPEC_ERROR	14	X'0000000E'
MQRQ_SSL_CLIENT_AUTH_ERROR	15	X'0000000F'
MQRQ_SSL_PEER_NAME_ERROR	16	X'00000010'
MQRQ_SUB_NOT_AUTHORIZED	17	X'00000011'
MQRQ_SUB_DEST_NOT_AUTHORIZED	18	X'00000012'
MQRQ_SSL_UNKNOWN_REVOCATION	19	X'00000013'
MQRQ_SYS_CONN_NOT_AUTHORIZED	20	X'00000014'
MQRQ_CHANNEL_BLOCKED_ADDRESS	21	X'00000015'
MQRQ_CHANNEL_BLOCKED_USERID	22	X'00000016'
MQRQ_CHANNEL_BLOCKED_NOACCESS	23	X'00000017'
MQRQ_MAX_ACTIVE_CHANNELS	24	X'00000018'
MQRQ_MAX_CHANNELS	25	X'00000019'

<i>Tabela 327. Valores de constantes (continuação)</i>		
Nome	Valor decimal	Valor Hexadecimal
MQRQ_SVRCONN_INST_LIMIT	26	X'0000001A'
MQRQ_CLIENT_INST_LIMIT!	27	X'0000001B'
MQRQ_CAF_NOT_INSTALADO	28	X'0000001C'
MQRQ_CSP_NOT_AUTHORIZED	29	X'0000001D'
MQRQ_FAILOVER_PERMITIDO	30	X'0000001E'
MQRQ_FAILOVER_NOT_PERMITIDO	31	X'0000001F'
MQRQ_STANDBY_ACTIVATED	32	X'00000020'
MQRQ_REPLICA_ATIVADO	33	X'00000021'

### **MQRT\_\* (tipos de atualização de formato de comando)**

<i>Tabela 328. Valores de constantes</i>		
Nome	Valor decimal	Valor Hexadecimal
MQRT_CONFIGURATION	1	X'00000001'
MQRT_EXPIRE	2	X'00000002'
MQRT_NSPROC	3	X'00000003'
MQRT_PROXYSUB	4	X'00000004'

### **MQRU\_\* (Somente solicitação)**

<i>Tabela 329. Valores de constantes</i>		
Nome	Valor decimal	Valor Hexadecimal
MQRU_PUBLISH_ON_REQUEST	1	X'00000001'
MQRU_PUBLISH_ALL	2	X'00000002'

### **MQSCA\_\* (Autenticação de Cliente TLS).**

<i>Tabela 330. Valores de constantes</i>		
Nome	Valor decimal	Valor Hexadecimal
MQSCA_REQUIRED	0	X'00000000'
MQSCA_OPTIONAL	1	X'00000001'

### **MQSCO\_\* (opções de configuração TLS)**

#### **Estrutura de opções de configuração TLS**

<i>Tabela 331. Estruturas de constantes</i>	
Nome	Estrutura
MQSCO_STRUC_ID	"SCO~"
MQSCO_STRUC_ID_ARRAY	'S','C','O','~'

**Nota:** O símbolo ~ representa um único caractere em branco.

Tabela 332. Valores de constantes		
Nome	Valor decimal	Valor Hexadecimal
MQSCO_VERSION_1	1	X'00000001'
MQSCO_VERSION_2	2	X'00000002'
MQSCO_VERSION_3	3	X'00000003'
MQSCO_VERSION_4	4	X'00000004'
MQSCO_CURRENT_VERSION	4	X'00000004'

**Nota:** O símbolo – representa um único caractere em branco.

### Contagem de reconfiguração de chave de opções de configuração TLS

Tabela 333. Valores de constantes		
Nome	Valor decimal	Valor Hexadecimal
MQSCO_RESET_COUNT_DEFAULT	0	X'00000000'

### Escopo de Definição de Fila do formato de comando....

Tabela 334. Valores de constantes		
Nome	Valor decimal	Valor Hexadecimal
MQSCO_Q_MGR	1	X'00000001'
MQSCO_CELL	2	X'00000002'

### MQSCOPE\_\* (escopo de publicação)

Tabela 335. Valores de constantes		
Nome	Valor decimal	Valor Hexadecimal
MQSCOPE_ALL	0	X'00000000'
MQSCOPE_AS_PARENT	1	X'00000001'
MQSCOPE_QMGR	4	X'00000004'

### MQSCYC\_\* (Caso de Segurança)

Tabela 336. Valores de constantes		
Nome	Valor decimal	Valor Hexadecimal
MQSCYC_UPPER	0	X'00000000'
MQSCYC_MIXED	1	X'00000001'

### MQSD\_\* (Estrutura do descritor de objeto)

Tabela 337. Nomes e estruturas de constantes	
Nome	Estrutura
MQSD_STRUC_ID	"SD--"
MQSD_STRUC_ID_ARRAY	'S','D','-',','

**Nota:** O símbolo – representa um único caractere em branco.

<i>Tabela 338. Valores de constantes</i>		
Nome	Valor decimal	Valor Hexadecimal
MQSD_VERSION_1	1	X'00000001'
MQSD_CURRENT_VERSION	1	X'00000001'

### **MQSECITEM\_\* (Itens de Segurança do Formato de Comandos)..**

<i>Tabela 339. Valores de constantes</i>		
Nome	Valor decimal	Valor Hexadecimal
MQSECITEM_ALL	0	X'00000000'
MQSECITEM_MQADMIN	1	X'00000001'
MQSECITEM_MQNLIST	2	X'00000002'
MQSECITEM_MQPROC	3	X'00000003'
MQSECITEM_MQQUEUE	4	X'00000004'
MQSECITEM_MQCONN	5	X'00000005'
MQSECITEM_MQCMDS	6	X'00000006'
MQSECITEM_MXADMIN	7	X'00000007'
MQSECITEM_MXNLIST	8	X'00000008'
MQSECITEM_MXPROC	9	X'00000009'
MQSECITEM_MXQUEUE	10	X'0000000A'
MQSECITEM_MXTOPIC	11	X'0000000B'

### **MQSECPROT\_\* (Tipos de Protocolo de Segurança)**

<i>Tabela 340. Valores de constantes</i>		
Nome	Valor decimal	Valor Hexadecimal
MQSECPROT_NONE	0	X'00000000'
MQSECPROT_SSLV30	1	X'00000001'
MQSECPROT_TL SV10	2	X'00000002'
MQSECPROT_TL SV12	4	X'00000004'

### **MQSECSW\_\* (Comutadores de Segurança e Estados do Computador de formato de comando)..**

#### **Comutadores de Segurança de Formato de Comando**

<i>Tabela 341. Valores de constantes</i>		
Nome	Valor decimal	Valor Hexadecimal
MQSECSW_PROCESS	1	X'00000001'
MQSECSW_NAMELIST	2	X'00000002'
MQSECSW_Q	3	X'00000003'
MQSECSW_TOPIC	4	X'00000004'
MQSECSW_CONTEXT	6	X'00000006'
MQSECSW_ALTERNATE_USER	7	X'00000007'
MQSECSW_COMMAND	8	X'00000008'



*Tabela 341. Valores de constantes (continuação)*

Nome	Valor decimal	Valor Hexadecimal
MQSECSW_CONNECTION	9	X'00000009'
MQSECSW_SUBSYSTEM	10	X'0000000A'
MQSECSW_COMMAND_RESOURCES	11	X'0000000B'
MQSECSW_Q_MGR	15	X'0000000F'
MQSECSW_QSG	16	X'00000010'

### Estados do Computador de Segurança

*Tabela 342. Valores de constantes*

Nome	Valor decimal	Valor Hexadecimal
MQSECSW_OFF_FOUND	21	X'00000015'
MQSECSW_ON_FOUND	22	X'00000016'
MQSECSW_OFF_NOT_FOUND	23	X'00000017'
MQSECSW_ON_NOT_FOUND	24	X'00000018'
MQSECSW_OFF_ERROR	25	X'00000019'
MQSECSW_ON_OVERRIDDEN	26	X'0000001A'

### MQSECTYPE\_\* (Tipos de Segurança do Formato de Comando)..

*Tabela 343. Valores de constantes*

Nome	Valor decimal	Valor Hexadecimal
MQSECTYPE_AUTHSERV	1	X'00000001'
MQSECTYPE_SSL	2	X'00000002'
MQSECTYPE_CLASSES	3	X'00000003'

### MQSEG\_\* (Segmentação)

*Tabela 344. Nomes e valores de constantes*

Nome	Value
MQSEG_INIBIDA	'↵'
MQSEG_ALLOWED	'A'

**Nota:** O símbolo ↵ representa um único caractere em branco.

### MQSEL\_\* (Valores do Seletor Especial).

*Tabela 345. Valores de constantes*

Nome	Valor decimal	Valor Hexadecimal
MQSEL_ANY_SELECTOR	-30001	X'FFFF8ACF'
MQSEL_ANY_USER_SELECTOR	-30002	X'FFFF8ACE'
MQSEL_ANY_SYSTEM_SELECTOR	-30003	X'FFFF8ACD'
MQSEL_ALL_SELECTORS	-30001	X'FFFF8ACF'
MQSEL_ALL_USER_SELECTORS	-30002	X'FFFF8ACE'
MQSEL_ALL_SYSTEM_SELECTORS	-30003	X'FFFF8ACD'

## MQSELTYPE\_\* (Tipos de Seletor).

Nome	Valor decimal	Valor Hexadecimal
MQSELTYPE_NONE	0	X'00000000'
MQSELTYPE_STANDARD	1	X'00000001'
MQSELTYPE_EXTENDED	2	X'00000002'

## MQSID\_\* (Identificador de Segurança)

Nome	Value
MQSID_NONE	X'00...00' (40 nulos)
MQSID_NONE_ARRAY	'\0', '\0', ... (40 nulos)

## MQSIDT\_\* (Tipos de Identificador de Segurança)

Nome	Valor Hexadecimal
MQSIDT_NONE	X'00'
MQSIDT_NT_SECURITY_ID	X'01'
MQSIDT_WAS_SECURITY_ID	X'02'

## MQSMPO\_\* (Configurar opções de propriedade de mensagem e estrutura).

### Configurar estrutura de opções de propriedade de mensagem

Nome	Estrutura
MQSMPO_STRUC_ID	"SMPO"
MQSMPO_STRUC_ID_ARRAY	'S', 'M', 'P', 'O'

**Nota:** O símbolo ~ representa um único caractere em branco.

Nome	Valor decimal	Valor Hexadecimal
MQSMPO_VERSION_1	1	X'00000001'
MQSMPO_CURRENT_VERSION	1	X'00000001'

### Configurar opções de propriedade de mensagem

Nome	Valor decimal	Valor Hexadecimal
MQSMPO_SET_FIRST	0	X'00000000'
MQSMPO_SET_PROP_UNDER_CURSOR	1	X'00000001'
MQSMPO_SET_PROP_AFTER_CURSOR	2	X'00000002'
MQSMPO_APPEND_PROPERTY	4	X'00000004'
MQSMPO_SET_PROP_BEFORE_CURSOR	8	X'00000008'

Tabela 351. Valores de constantes (continuação)

Nome	Valor decimal	Valor Hexadecimal
MQSMPO_NONE	0	X'00000000'

### MQSO\_\* (Opções de Assinatura)

Tabela 352. Valores de constantes

Nome	Valor decimal	Valor Hexadecimal
MQSO_NONE	0	X'00000000'
MQSO_NON_DURABLE	0	X'00000000'
MQSO_READ_AHEAD_AS_Q_DEF	0	X'00000000'
MQSO_ALTER	1	X'00000001'
MQSO_CREATE	2	X'00000002'
MQSO_RESUME	4	X'00000004'
MQSO_DURABLE	8	X'00000008'
MQSO_GROUP_SUB	16	X'00000010'
MQSO_MANAGED	32	X'00000020'
MQSO_SET_IDENTITY_CONTEXT	64	X'00000040'
MQSO_FIXED_USERID	256	X'00000100'
MQSO_ANY_USERID	512	X'00000200'
MQSO_PUBLICATIONS_ON_REQUEST	2048	X'00000800'
MQSO_NEW_PUBLICATIONS_ONLY	4096	X'00001000'
MQSO_FAIL_IF QUIESCING	8192	X'00002000'
MQSO_ALTERNATE_USER_AUTHORITY	262144	X'00040000'
MQSO_WILDCARD_CHAR	1048576	X'00100000'
MQSO_WILDCARD_TOPIC	2097152	X'00200000'
MQSO_SET_CORREL_ID	4194304	X'00400000'
MQSO_SCOPE_QMGR	67108864	X'04000000'
MQSO_NO_READ_AHEAD	134217728	X'08000000'
MQSO_READ_AHEAD	268435456	X'10000000'

### MQSP\_\* (Disponibilidade do ponto de sincronização)

Tabela 353. Valores de constantes

Nome	Valor decimal	Valor Hexadecimal
MQSP_AVAILABLE	1	X'00000001'
MQSP_NOT_AVAILABLE	0	X'00000000'

### MQSPL\_\* (Opções de proteção de política de segurança)

Tabela 354. Valores de constantes

Nome	Valor decimal	Valor Hexadecimal
MQSPL_PASSTHRU	0	X'00000000'
MQSPL_REMOVE	1	X'00000001'
MQSPL_AS_POLICY	2	X'00000002'

## MQSQQM\_\* (Nome do Gerenciador de Filas de Fila Compartilhada)

Tabela 355. Valores de constantes		
Nome	Valor decimal	Valor Hexadecimal
MQSQQM_USE	0	X'00000000'
MQSQQM_IGNORE	1	X'00000001'

## MQSR\_\* (Ação).

Tabela 356. Valores de constantes		
Nome	Valor decimal	Valor Hexadecimal
MQSR_ACTION_PUBLICATION	1	X'00000001'

## MQSRO\_\* (Estrutura de opções de solicitação de assinatura)

Tabela 357. Estruturas de constantes	
Nome	Estrutura
MQSRO_STRUC_ID	"SRO↵"
MQSRO_STRUC_ID_ARRAY	'S', 'R', 'O', '↵'

**Nota:** O símbolo ↵ representa um único caractere em branco.

Tabela 358. Valores de constantes		
Nome	Valor decimal	Valor Hexadecimal
MQSRO_VERSION_1	1	X'00000001'
MQSRO_CURRENT_VERSION	1	X'00000001'
MQSRO_NONE	0	X'00000000'
MQSRO_FAIL_IF_QUIESCING	8192	X'00002000'

## MQSS\_\* (Status do segmento)

Tabela 359. Nomes e estruturas de constantes	
Nome	Estrutura
MQSS_NOT_A_SEGMENT	'↵'
MQSS_SEGMENT	'S'
MQSS_LAST_SEGMENT	'L'

**Nota:** O símbolo ↵ representa um único caractere em branco.

## MQSSL\_\* (Requisitos do TLS FIPS)

**Nota:** No AIX, Linux, and Windows, IBM MQ fornece conformidade FIPS 140-2 por meio do módulo criptográfico IBM Crypto for C (ICC) . O certificado deste módulo foi movido para o status Histórico. Os clientes devem visualizar o [IBM Crypto for C \(ICC\) certificado](#) e estar ciente de qualquer aviso fornecido pelo NIST Um módulo FIPS 140-3 de substituição está atualmente em andamento e seu status pode ser visualizado procurando por ele na [NIST CMVP modules in process list](#).

Tabela 360. Valores de constantes		
Nome	Valor decimal	Valor Hexadecimal
MQSSL_FIPS_NO	0	X'00000000'

Tabela 360. Valores de constantes (continuação)		
Nome	Valor decimal	Valor Hexadecimal
MQSSL_FIPS_YES	1	X'00000001'

### MQSTAT\_\* (Opções de Stat)

Tabela 361. Valores de constantes		
Nome	Valor decimal	Valor Hexadecimal
MQSTAT_TYPE_ASYNC_ERROR	0	X'00000000'
MQSTAT_TYPE_RECONNECTION	0	X'00000000'
MQSTAT_TYPE_RECONNECTION_ERROR	0	X'00000000'

### MQSTS\_\* (estrutura de relatório de status)

Tabela 362. Estruturas de constantes	
Nome	Estrutura
MQSTS_STRUC_ID	"STAT"
MQSTS_STRUC_ID_ARRAY	'S', 'T', 'A', 'T'

**Nota:** O símbolo ~ representa um único caractere em branco.

Tabela 363. Valores de constantes		
Nome	Valor decimal	Valor Hexadecimal
MQSTS_VERSION_1	1	X'00000001'
MQSTS_CURRENT_VERSION	1	X'00000001'

### MQSUB\_\* (assinaturas duráveis)

#### Assinaturas permitidas duráveis

Tabela 364. Valores de constantes		
Nome	Valor decimal	Valor Hexadecimal
MQSUB_DURABLE_AS_PARENT	0	X'00000000'
MQSUB_DURABLE_ALLOWED	1	X'00000001'
MQSUB_DURABLE_INIBIDA	2	X'00000002'

#### Intervalo de assinaturas duráveis

Tabela 365. Valores de constantes		
Nome	Valor decimal	Valor Hexadecimal
MQSUB_DURABLE_ALL	-1	X'FFFFFFFF'
MQSUB_DURABLE_YES	1	X'00000001'
MQSUB_DURABLE_NO	2	X'00000002'

## MQSUBTYPE\_\* (Tipos de Assinatura de Formato de Comandos).

Tabela 366. Valores de constantes		
Nome	Valor decimal	Valor Hexadecimal
MQSUBTYPE_API	1	X'00000001'
MQSUBTYPE_ADMIN	2	X'00000002'
MQSUBTYPE_PROXY	3	X'00000003'
MQSUBTYPE_ALL	-1	X'FFFFFFFF'
MQSUBTYPE_USER	-2	X'FFFFFFFE'

## MQSUS\_\* (Formato de Comando Suspend Status)

Tabela 367. Valores de constantes		
Nome	Valor decimal	Valor Hexadecimal
MQSUS_YES	1	X'00000001'
MQSUS_NO	0	X'00000000'

## MQSVC\_\* (Serviço)

### Tipos de serviço

Tabela 368. Valores de constantes		
Nome	Valor decimal	Valor Hexadecimal
MQSVC_TYPE_COMMAND	0	X'00000000'
MQSVC_TYPE_SERVER	1	X'00000001'

### Controles de Serviço.

Tabela 369. Valores de constantes		
Nome	Valor decimal	Valor Hexadecimal
MQSVC_CONTROL_Q_MGR	0	X'00000000'
MQSVC_CONTROL_Q_MGR_START	1	X'00000001'
MQSVC_CONTROL_MANUAL	2	X'00000002'

### Status de Serviço

Tabela 370. Valores de constantes		
Nome	Valor decimal	Valor Hexadecimal
MQSVC_STATUS_STOPPED	0	X'00000000'
MQSVC_STATUS_STARTING	1	X'00000001'
MQSVC_STATUS_RUNNING	2	X'00000002'
MQSVC_STATUS_STOPPING	3	X'00000003'
MQSVC_STATUS_RETRYING	4	X'00000004'

## MQSYNCPPOINT\_\* (Valores de ponto de sincronização do formato de comando para migração de Pub / Sub)

*Tabela 371. Valores de constantes*

Nome	Valor decimal	Valor Hexadecimal
MQSYNCPPOINT_YES	0	X'00000000'
MQSYNCPPOINT_IFPER	1	X'00000001'

## MQSYSP\_\* (Valores de parâmetro do sistema de formato de comando)

*Tabela 372. Valores de constantes*

Nome	Valor decimal	Valor Hexadecimal
MQSYSP_NO	0	X'00000000'
MQSYSP_YES	1	X'00000001'
MQSYSP_EXTENDED	2	X'00000002'
MQSYSP_TYPE_INITIAL	10	X'0000000A'
MQSYSP_TYPE_SET	11	X'0000000B'
MQSYSP_TYPE_LOG_COPY	12	X'0000000C'
MQSYSP_TYPE_LOG_STATUS	13	X'0000000D'
MQSYSP_TYPE_ARCHIVE_TAPE	14	X'0000000E'
MQSYSP_ALLOC_BLK	20	X'00000014'
MQSYSP_ALLOC_TRK	21	X'00000015'
MQSYSP_ALLOC_CYL	22	X'00000016'
MQSYSP_STATUS_BUSY	30	X'0000001E'
MQSYSP_STATUS_PREMOUNT	31	X'0000001F'
MQSYSP_STATUS_AVAILABLE	32	X'00000020'
MQSYSP_STATUS_UNKNOWN	33	X'00000021'
MQSYSP_STATUS_ALLOC_ARCHIVE	34	X'00000022'
MQSYSP_STATUS_COPYING_BSDS	35	X'00000023'
MQSYSP_STATUS_COPYING_LOG	36	X'00000024'

## MQTA\_\* (Atributos de Tópico).

### Curingas

*Tabela 373. Valores de constantes*

Nome	Valor decimal	Valor Hexadecimal
MQTA_BLOCK	1	X'00000001'
MQTA_PASSTHRU	2	X'00000002'

### Assinaturas permitidas

*Tabela 374. Valores de constantes*

Nome	Valor decimal	Valor Hexadecimal
MQTA_SUB_AS_PARENT	0	X'00000000'

<i>Tabela 374. Valores de constantes (continuação)</i>		
Nome	Valor decimal	Valor Hexadecimal
MQTA_SUB_INIBIDA	1	X'00000001'
MQTA_SUB_ALLOWED	2	X'00000002'

### Subpropagação de Proxy

<i>Tabela 375. Valores de constantes</i>		
Nome	Valor decimal	Valor Hexadecimal
MQTA_PROXY_SUB_FORCE	1	X'00000001'
MQTA_PROXY_SUB_FIRSTUSE	2	X'00000002'

### Publicações permitidas

<i>Tabela 376. Valores de constantes</i>		
Nome	Valor decimal	Valor Hexadecimal
MQTA_PUB_AS_PARENT	0	X'00000000'
MQTA_PUB_INIBIDA	1	X'00000001'
MQTA_PUB_ALLOWED	2	X'00000002'

### MQTC\_\* (Controles do acionador)

<i>Tabela 377. Valores de constantes</i>		
Nome	Valor decimal	Valor Hexadecimal
MQTC_OFF	0	X'00000000'
MQTC_ON	1	X'00000001'

### MQTCPKEEP\_\* (TCP Keepalive)

<i>Tabela 378. Valores de constantes</i>		
Nome	Valor decimal	Valor Hexadecimal
MQTCPKEEP_NO	0	X'00000000'
MQTCPKEEP_YES	1	X'00000001'

### MQTCPSTACK\_\* (Tipos de Pilhas TCP)

<i>Tabela 379. Valores de constantes</i>		
Nome	Valor decimal	Valor Hexadecimal
MQTCPSTACK_SINGLE	0	X'00000000'
MQTCPSTACK_MULTIPLE	1	X'00000001'

### MQTIME\_\* (Unidades de tempo de formato de comando)

<i>Tabela 380. Valores de constantes</i>		
Nome	Valor decimal	Valor Hexadecimal
MQTIME_UNIT_MINS	0	X'00000000'
MQTIME_UNIT_SECS	1	X'00000001'



## MQTM\_\* (Estrutura da mensagem do acionador)

Tabela 381. Estruturas de constantes	
Nome	Estrutura
MQTM_STRUC_ID	"TM--"
MQTM_STRUC_ID_ARRAY	'T','M',' ',' '

**Nota:** O símbolo - representa um único caractere em branco.

Tabela 382. Valores de constantes		
Nome	Valor decimal	Valor Hexadecimal
MQTM_VERSION_1	1	X'00000001'
MQTM_CURRENT_VERSION	1	X'00000001'

## MQTMC\_\* (Estrutura do formato de caractere da mensagem do acionador).

Tabela 383. Estruturas de constantes	
Nome	Estrutura
MQTMC_STRUC_ID	"TMC--"
MQTMC_STRUC_ID_ARRAY	'T','M','C',' ',' '
MQTMC_VERSION_1	"--1"
MQTMC_VERSION_2	"--2"
MQTMC_CURRENT_VERSION	"--2"
MQTMC_VERSION_1_ARRAY	' ',' ',' ','1'
MQTMC_VERSION_2_ARRAY	' ',' ',' ','2'
MQTMC_CURRENT_VERSION_ARRAY	' ',' ',' ','2'

## MQTOPT\_\* (Tipo de Tópico).

Tabela 384. Valores de constantes		
Nome	Valor decimal	Valor Hexadecimal
MQTOPT_LOCAL	0	X'00000000'
MQTOPT_CLUSTER	1	X'00000001'
MQTOPT_ALL	2	X'00000002'

## MQTRAXSTR\_\* (Autoinicialização do Rastreo do Inicializador de Canais)

Tabela 385. Valores de constantes		
Nome	Valor decimal	Valor Hexadecimal
MQTRAXSTR_NO	0	X'00000000'
MQTRAXSTR_YES	1	X'00000001'

## MQTSCOPE\_\* (Escopo da assinatura)

Tabela 386. Valores de constantes		
Nome	Valor decimal	Valor Hexadecimal
MQTSCOPE_QMGR	1	X'00000001'
MQTSCOPE_ALL	2	X'00000002'

## MQTT\_\* (Tipos de Acionador)

Tabela 387. Valores de constantes		
Nome	Valor decimal	Valor Hexadecimal
MQTT_NONE	0	X'00000000'
MQTT_FIRST	1	X'00000001'
MQTT EVERY	2	X'00000002'
MQTT_DEPTH	3	X'00000003'

## MQTYPE\_\* (Tipos de dados da propriedade).

Tabela 388. Valores de constantes		
Nome	Valor decimal	Valor Hexadecimal
MQTYPE_AS_SET	0	X'00000000'
MQTYPE_NULL	2	X'00000002'
MQTYPE_BOOLEAN	4	X'00000004'
MQTYPE_BYTE_STRING	8	X'00000008'
MQTYPE_INT8	16	X'00000010'
MQTYPE_INT16	32	X'00000020'
MQTYPE_INT32	64	X'00000040'
MQTYPE_LONG	64	X'00000040'
MQTYPE_INT64	128	X'00000080'
MQTYPE_FLOAT32	256	X'00000100'
MQTYPE_FLOAT64	512	X'00000200'
MQTYPE_STRING	1024	X'00000400'

## MQUA\_\* (Seletores de Atributo do Usuário de Publicação / Assinatura)

Tabela 389. Valores de constantes		
Nome	Valor decimal	Valor Hexadecimal
MQUA_FIRST	65536	X'00010000'
MQUA_LAST	99999999	X'3B9AC9FF'

## MQUIDSUPP\_\* (Suporte do ID do Usuário do formato de comandos)

Tabela 390. Valores de constantes		
Nome	Valor decimal	Valor Hexadecimal
MQUIDSUPP_NO	0	X'00000000'
MQUIDSUPP_YES	1	X'00000001'

## MQUDELIVERED\_\* (Valores não entregues no formato de comando para migração de Pub / Sub);

Tabela 391. Valores de constantes		
Nome	Valor decimal	Valor Hexadecimal
MQUDELIVERED_NORMAL	0	X'00000000'

Tabela 391. Valores de constantes (continuação)		
Nome	Valor decimal	Valor Hexadecimal
MQUNDELIVERED_SAFE	1	X'00000001'
MQUNDELIVERED_DISCARD	2	X'00000002'
MQUNDELIVERED_KEEP	3	X'00000003'

### MQUOWST\_\* (Estados da UOW do formato de comando).

Tabela 392. Valores de constantes		
Nome	Valor decimal	Valor Hexadecimal
MQUOWST_NONE	0	X'00000000'
MQUOWST_ACTIVE	1	X'00000001'
MQUOWST_PREPARED	2	X'00000002'
MQUOWST_UNRESOLVED	3	X'00000003'

### MQUOWT\_\* (Tipos de UOW no formato de Comando).

Tabela 393. Valores de constantes		
Nome	Valor decimal	Valor Hexadecimal
MQUOWT_Q_MGR	0	X'00000000'
MQUOWT_CICS	1	X'00000001'
MQUOWT_RRS	2	X'00000002'
MQUOWT_IMS	3	X'00000003'
MQUOWT_XA	4	X'00000004'

### MQUS\_\* (Usos de fila).

Tabela 394. Valores de constantes		
Nome	Valor decimal	Valor Hexadecimal
MQUS_NORMAL	0	X'00000000'
MQUS_TRANSMISSION	1	X'00000001'

### MQUSAGE\_\* (Valores de Uso do Conjunto de Páginas e Valores de Uso do Conjunto de Dados)

#### Valores de Uso do Conjunto de Páginas do formato de comando

Tabela 395. Valores de constantes		
Nome	Valor decimal	Valor Hexadecimal
MQUSAGE_PS_AVAILABLE	0	X'00000000'
MQUSAGE_PS_DEFINED	1	X'00000001'
MQUSAGE_PS_OFFLINE	2	X'00000002'
MQUSAGE_PS_NOT_DEFINED	3	X'00000003'
MQUSAGE_PS_SUSPENDED	4	X'00000004'
MQUSAGE_EXPAND_USER	1	X'00000001'
MQUSAGE_EXPAND_SYSTEM	2	X'00000002'

<i>Tabela 395. Valores de constantes (continuação)</i>		
Nome	Valor decimal	Valor Hexadecimal
MQUSAGE_EXPAND_NONE	3	X'00000003'

### Valores de Uso do Conjunto de Dados do Formato de Comando.

<i>Tabela 396. Valores de constantes</i>		
Nome	Valor decimal	Valor Hexadecimal
MQUSAGE_DS_OLDEST_ACTIVE_UOW	10	X'0000000A'
MQUSAGE_DS_OLDEST_PS_RECOVERY	11	X'0000000B'
MQUSAGE_DS_OLDEST_CF_RECOVERY	12	X'0000000C'

### MQVL\_\* (comprimento do valor)

<i>Tabela 397. Valores de constantes</i>		
Nome	Valor decimal	Valor Hexadecimal
MQVL_NULL_TERMINATED	-1	X'FFFFFFFF'
MQVL_EMPTY_STRING	0	X'00000000'

### MQVU\_\* (ID do Usuário Variável)

<i>Tabela 398. Valores de constantes</i>		
Nome	Valor decimal	Valor Hexadecimal
MQVU_FIXED_USER	1	X'00000001'
MQVU_ANY_USER	2	X'00000002'

### MQWDR\_\* (Estrutura de registro de destino de saída de carga de trabalho do cluster)

<i>Tabela 399. Estruturas de constantes</i>	
Nome	Estrutura
MQWDR_STRUC_ID	"WDR↵"
MQWDR_STRUC_ID_ARRAY	'W', 'D', 'R', '↵'

**Nota:** O símbolo ↵ representa um único caractere em branco.

<i>Tabela 400. Valores de constantes</i>		
Nome	Valor decimal	Valor Hexadecimal
MQWDR_VERSION_1	1	X'00000001'
MQWDR_VERSION_2	2	X'00000002'
MQWDR_CURRENT_VERSION	2	X'00000002'
MQWDR_LENGTH_1	124	X'0000007C'
MQWDR_LENGTH_2	136	X'00000088'
MQWDR_CURRENT_LENGTH	136	X'00000088'

## MQWI\_\* (Intervalo de Espera)

Tabela 401. Valores de constantes		
Nome	Valor decimal	Valor Hexadecimal
MQWI_UNLIMITED	-1	X'FFFFFFFF'

## MQWIH\_\* (Estrutura do cabeçalho de informações de carga de trabalho e Sinalizadores)

### Estrutura do cabeçalho de informações de carga de trabalho

Tabela 402. Estruturas de constantes	
Nome	Estrutura
MQWIH_STRUC_ID	"WIH~"
MQWIH_STRUC_ID_ARRAY	'W', 'I', 'H', '~'

**Nota:** O símbolo ~ representa um único caractere em branco.

Tabela 403. Valores de constantes		
Nome	Valor decimal	Valor Hexadecimal
MQWIH_VERSION_1	1	X'00000001'
MQWIH_CURRENT_VERSION	1	X'00000001'
MQWIH_LENGTH_1	120	X'00000078'
MQWIH_CURRENT_LENGTH	120	X'00000078'

### Sinalizadores do cabeçalho de informações de carga de trabalho

Tabela 404. Valores de constantes		
Nome	Valor decimal	Valor Hexadecimal
MQWIH_NONE	0	X'00000000'

## MQWQR\_\* (Estrutura de registro da fila de saída da carga de trabalho do cluster).

Tabela 405. Estruturas de constantes	
Nome	Estrutura
MQWQR_STRUC_ID	"WQR~"
MQWQR_STRUC_ID_ARRAY	'W', 'Q', 'R', '~'

**Nota:** O símbolo ~ representa um único caractere em branco.

Tabela 406. Valores de constantes		
Nome	Valor decimal	Valor Hexadecimal
MQWQR_VERSION_1	1	X'00000001'
MQWQR_VERSION_2	2	X'00000002'
MQWQR_VERSION_3	3	X'00000003'
MQWQR_CURRENT_VERSION	3	X'00000003'
MQWQR_LENGTH_1	200	X'000000C8'

<i>Tabela 406. Valores de constantes (continuação)</i>		
Nome	Valor decimal	Valor Hexadecimal
MQWQR_LENGTH_2	208	X'000000D0'
MQWQR_LENGTH_3	212	X'000000D4'
MQWQR_CURRENT_LENGTH	212	X'000000D4'

### **MQWS\_\* (Esquema Curinga)**

<i>Tabela 407. Valores de constantes</i>		
Nome	Valor decimal	Valor Hexadecimal
MQWS_DEFAULT	0	X'00000000'
MQWS_CHAR	1	X'00000001'
MQWS_TOPIC	2	X'00000002'

### **MQWXP\_\* (Estrutura do parâmetro de saída de carga de trabalho do cluster)**

### **MQWXP\_\* (Estrutura do parâmetro de saída de carga de trabalho do cluster)**

<i>Tabela 408. Estruturas de constantes</i>	
Nome	Estrutura
MQWXP_STRUC_ID	"WXP¬"
MQWXP_STRUC_ID_ARRAY	'W', 'X', 'P', '¬'

**Nota:** O símbolo ¬ representa um único caractere em branco.

<i>Tabela 409. Valores de constantes</i>		
Nome	Valor decimal	Valor Hexadecimal
MQWXP_VERSION_1	1	X'00000001'
MQWXP_VERSION_2	2	X'00000002'
MQWXP_VERSION_3	3	X'00000003'
MQWXP_VERSION_4	4	X'00000004'
MQWXP_CURRENT_VERSION	4	X'00000004'

### **MQWXP\_\* (Sinalizadores de carga de trabalho do cluster)**

<i>Tabela 410. Valores de constantes</i>		
Nome	Valor decimal	Valor Hexadecimal
MQWXP_PUT_BY_CLUSTER_CHL	2	X'00000002'

#### **Referências relacionadas**

“Campos em MQWXP -Estrutura do parâmetro de saída de carga do cluster” na página 1598  
 Descrição dos campos na estrutura do parâmetro de saída de carga de trabalho MQWXP -Cluster

### **MQXACT\_\* (Tipos de Chamada da API)**

<i>Tabela 411. Valores de constantes</i>		
Nome	Valor decimal	Valor Hexadecimal
MQXACT_EXTERNAL	1	X'00000001'

<i>Tabela 411. Valores de constantes (continuação)</i>		
<b>Nome</b>	<b>Valor decimal</b>	<b>Valor Hexadecimal</b>
MQXACT_INTERNAL	2	X'00000002'

### **MQXC\_\* (Comandos de Saída)**

<i>Tabela 412. Valores de constantes</i>		
<b>Nome</b>	<b>Valor decimal</b>	<b>Valor Hexadecimal</b>
MQXC_MQOPEN	1	X'00000001'
MQXC_MQCLOSE	2	X'00000002'
MQXC_MQGET	3	X'00000003'
MQXC_MQPUT	4	X'00000004'
MQXC_MQPUT1	5	X'00000005'
MQXC_MQINQ	6	X'00000006'
MQXC_MQSET	8	X'00000008'
MQXC_MQBACK	9	X'00000009'
MQXC_MQCMIT	10	X'0000000A'

### **MQXCC\_\* (respostas de saída)**

<i>Tabela 413. Valores de constantes</i>		
<b>Nome</b>	<b>Valor decimal</b>	<b>Valor Hexadecimal</b>
MQXCC_OK	0	X'00000000'
MQXCC_SUPPRESS_FUNCTION	-1	X'FFFFFFFF'
MQXCC_SKIP_FUNCTION	-2	X'FFFFFFFE'
MQXCC_SEND_AND_REQUEST_SEC_MSG	-3	X'FFFFFFFD'
MQXCC_SEND_SEC_MSG	-4	X'FFFFFFFC'
MQXCC_SUPPRESS_EXIT	-5	X'FFFFFFFB'
MQXCC_CLOSE_CHANNEL	-6	X'FFFFFFFA'
MQXCC_REQUEST_ACK	-7	X'FFFFFFF9'
MQXCC_FAILED	-8	X'FFFFFFF8'

### **MQXDR\_\* (Resposta de Saída)**

<i>Tabela 414. Valores de constantes</i>		
<b>Nome</b>	<b>Valor decimal</b>	<b>Valor Hexadecimal</b>
MQXDR_OK	0	X'00000000'
MQXDR_CONVERSION_FAILED	1	X'00000001'

### **MQXE\_\* (Ambientes)**

<i>Tabela 415. Valores de constantes</i>		
<b>Nome</b>	<b>Valor decimal</b>	<b>Valor Hexadecimal</b>
MQXE_OTHER	0	X'00000000'
MQXE_MCA	1	X'00000001'
MQXE_MCA_SVRCONN	2	X'00000002'

<i>Tabela 415. Valores de constantes (continuação)</i>		
Nome	Valor decimal	Valor Hexadecimal
MQXE_COMMAND_SERVER	3	X'00000003'
MQXE_MQSC	4	X'00000004'

## MQXEPO\_\* (Registrar a Estrutura de Opções do Ponto de Entrada e Opções de Saída).

### Estrutura de Opções do Ponto de Entrada de Registro.

<i>Tabela 416. Estruturas de constantes</i>	
Nome	Estrutura
MQXEPO_STRUC_ID	"XEPO"
MQXEPO_STRUC_ID_ARRAY	'X','E','P','O'

**Nota:** O símbolo ~ representa um único caractere em branco.

<i>Tabela 417. Valores de constantes</i>		
Nome	Valor decimal	Valor Hexadecimal
MQXEPO_VERSION_1	1	X'00000001'
MQXEPO_CURRENT_VERSION	1	X'00000001'

### Opções de saída.

<i>Tabela 418. Valores de constantes</i>		
Nome	Valor decimal	Valor Hexadecimal
MQXEPO_NONE	0	X'00000000'

## MQXF\_\* (Identificadores de função da API).

<i>Tabela 419. Valores de constantes</i>		
Nome	Valor decimal	Valor Hexadecimal
MQXF_INIT	1	X'00000001'
MQXF_TERM	2	X'00000002'
MQXF_CONN	3	X'00000003'
MQXF_CONNX	4	X'00000004'
MQXF_DISC	5	X'00000005'
MQXF_OPEN	6	X'00000006'
MQXF_CLOSE	7	X'00000007'
MQXF_PUT1	8	X'00000008'
MQXF_PUT	9	X'00000009'
MQXF_GET	10	X'0000000A'
MQXF_DATA_CONV_ON_GET	11	X'0000000B'
MQXF_INQ	12	X'0000000C'
MQXF_SET	13	X'0000000D'
MQXF_BEGIN	14	X'0000000E'



Tabela 419. Valores de constantes (continuação)		
Nome	Valor decimal	Valor Hexadecimal
MQXF_CMIT	15	X'0000000F'
MQXF_BACK	16	X'00000010'
MQXF_STAT	18	X'00000012'
MQXF_CB	19	X'00000013'
MQXF_CTL	20	X'00000014'
MQXF_CALLBACK	21	X'00000015'
MQXF_SUB	22	X'00000016'
MQXF_SUBRQ	23	X'00000017'
MQXF_XACLOSE	24	X'00000018'
MQXF_XACOMMIT	25	X'00000019'
MQXF_XACOMLETE	26	X'0000001A'
MQXF_XAEND	27	X'0000001B'
MQXF_XAFORGET	28	X'0000001C'
MQXF_XAOPEN	29	X'0000001D'
MQXF_XAPREPARE	30	X'0000001E'
MQXF_XARECOVER	31	X'0000001F'
MQXF_XAROLLBACK	32	X'00000020'
MQXF_XASTART	33	X'00000021'
MQXF_AXREG	34	X'00000022'
MQXF_AXUNREG	35	X'00000023'

### MQXP\_\* (estrutura de parâmetro de saída cruzada da API)

Tabela 420. Estruturas de constantes	
Nome	Estrutura
MQXP_STRUC_ID	"XP¬¬"
MQXP_STRUC_ID_ARRAY	'X', 'P', '¬', '¬'

**Nota:** O símbolo ¬ representa um único caractere em branco.

Tabela 421. Valores de constantes		
Nome	Valor decimal	Valor Hexadecimal
MQXP_VERSION_1	1	X'00000001'

### MQXPDA\_\* (Área de determinação de problema)

Tabela 422. Nomes e valores de constantes	
Nome	Value
MQXPDA_NONE	X'00...00' (48 nulos)
MQXPDA_NONE_ARRAY	'\0', '\0', ... (48 nulos)

## MQXPT\_\* (Tipos de Transporte).

*Tabela 423. Valores de constantes*

Nome	Valor decimal	Valor Hexadecimal
MQXPT_ALL	-1	X'FFFFFFFF'
MQXPT_LOCAL	0	X'00000000'
MQXPT_LU62	1	X'00000001'
MQXPT_TCP	2	X'00000002'
MQXPT_NETBIOS	3	X'00000003'
MQXPT_SPX	4	X'00000004'
MQXPT_DECNET	5	X'00000005'
MQXPT_UDP	6	X'00000006'

## MQXQH\_\* (Estrutura do cabeçalho da fila de transmissão)

*Tabela 424. Estruturas de constantes*

Nome	Estrutura
MQXQH_STRUC_ID	"XQH↵"
MQXQH_STRUC_ID_ARRAY	'X','Q','H','↵'

**Nota:** O símbolo ↵ representa um único caractere em branco.

*Tabela 425. Valores de constantes*

Nome	Valor decimal	Valor Hexadecimal
MQXQH_VERSION_1	1	X'00000001'
MQXQH_CURRENT_VERSION	1	X'00000001'

## MQXR\_\* (Motivos de saída).

*Tabela 426. Valores de constantes*

Nome	Valor decimal	Valor Hexadecimal
MQXR_BEFORE	1	X'00000001'
MQXR_AFTER	2	X'00000002'
MQXR_CONNECTION	3	X'00000003'
MQXR_INIT	11	X'0000000B'
MQXR_TERM	12	X'0000000C'
MQXR_MSG	13	X'0000000D'
MQXR_XMIT	14	X'0000000E'
MQXR_SEC_MSG	15	X'0000000F'
MQXR_INIT_SEC	16	X'00000010'
MQXR_RETRY	17	X'00000011'
MQXR_AUTO_CLUSSDR	18	X'00000012'
MQXR_AUTO_RECEIVER	19	X'00000013'
MQXR_CLWL_OPEN	20	X'00000014'
MQXR_CLWL_PUT	21	X'00000015'
MQXR_CLWL_MOVE	22	X'00000016'

<i>Tabela 426. Valores de constantes (continuação)</i>		
<b>Nome</b>	<b>Valor decimal</b>	<b>Valor Hexadecimal</b>
MQXR_CLWL_REPOS	23	X'00000017'
MQXR_CLWL_REPOS_MOVE	24	X'00000018'
MQXR_END_BATCH	25	X'00000019'
MQXR_ACK_RECEIVED	26	X'0000001A'
MQXR_AUTO_SVRCONN	27	X'0000001B'
MQXR_AUTO_CLUSRCVR	28	X'0000001C'
MQXR_SEC_PARS	29	X'0000001D'

### **MQXR2\_\* (Resposta de Saída 2)**

<i>Tabela 427. Valores de constantes</i>		
<b>Nome</b>	<b>Valor decimal</b>	<b>Valor Hexadecimal</b>
MQXR2_PUT_WITH_DEF_ACTION	0	X'00000000'
MQXR2_PUT_WITH_DEF_USERID	1	X'00000001'
MQXR2_PUT_WITH_MSG_USERID	2	X'00000002'
MQXR2_USE_AGENT_BUFFER	0	X'00000000'
MQXR2_USE_EXIT_BUFFER	4	X'00000004'
MQXR2_DEFAULT_CONTINUATION	0	X'00000000'
MQXR2_CONTINUE_CHAIN	8	X'00000008'
MQXR2_SUPPRESS_CHAIN	16	X'00000010'
MQXR2_STATIC_CACHE	0	X'00000000'
MQXR2_DYNAMIC_CACHE	32	X'00000020'

### **MQXT\_\* (Identificadores de Saída)**

<i>Tabela 428. Valores de constantes</i>		
<b>Nome</b>	<b>Valor decimal</b>	<b>Valor Hexadecimal</b>
MQXT_API_CROSSING_EXIT	1	X'00000001'
MQXT_API_EXIT	2	X'00000002'
MQXT_CHANNEL_SEC_EXIT	11	X'0000000B'
MQXT_CHANNEL_MSG_EXIT	12	X'0000000C'
MQXT_CHANNEL_SEND_EXIT	13	X'0000000D'
MQXT_CHANNEL_RCV_EXIT	14	X'0000000E'
MQXT_CHANNEL_MSG_RETRY_EXIT	15	X'0000000F'
MQXT_CHANNEL_AUTO_DEF_EXIT	16	X'00000010'
MQXT_CLUSTER_WORKLOAD_EXIT	20	X'00000014'
MQXT_PUBSUB_ROUTING_EXIT	21	X'00000015'

### **MQXUA\_\* (Valor da Área do Usuário de Saída)**

<i>Tabela 429. Nomes e valores de constantes</i>	
<b>Nome</b>	<b>Value</b>
MQXUA_NONE	X'00...00' (16 nulos)

Tabela 429. Nomes e valores de constantes (continuação)	
Nome	Value
MQXUA_NONE_ARRAY	'\0', '\0', ... (16 nulos)

### MQXWD\_\* (estrutura do descritor de espera de saída)

Tabela 430. Estruturas de constantes	
Nome	Estrutura
MQXWD_STRUC_ID	"XWD↵"
MQXWD_STRUC_ID_ARRAY	'X', 'W', 'D', '↵'

**Nota:** O símbolo ↵ representa um único caractere em branco.

Tabela 431. Valores de constantes		
Nome	Valor decimal	Valor Hexadecimal
MQXWD_VERSION_1	1	X'00000001'

### MQZAC\_\* (Estrutura de contexto de aplicativo)

Tabela 432. Estruturas de constantes	
Nome	Estrutura
MQZAC_STRUC_ID	"ZAC↵"
MQZAC_STRUC_ID_ARRAY	'Z', 'A', 'C', '↵'

**Nota:** O símbolo ↵ representa um único caractere em branco.

Tabela 433. Valores de constantes		
Nome	Valor decimal	Valor Hexadecimal
MQZAC_VERSION_1	1	X'00000001'
MQZAC_CURRENT_VERSION	1	X'00000001'

### MQZAD\_\* (Estrutura de dados de autoridade).

Tabela 434. Estruturas de constantes	
Nome	Estrutura
MQZAD_STRUC_ID	"ZAD↵"
MQZAD_STRUC_ID_ARRAY	'Z', 'A', 'D', '↵'

**Nota:** O símbolo ↵ representa um único caractere em branco.

Tabela 435. Valores de constantes		
Nome	Valor decimal	Valor Hexadecimal
MQZAD_VERSION_1	1	X'00000001'
MQZAD_VERSION_2	2	X'00000002'
MQZAD_CURRENT_VERSION	2	X'00000002'

## MQZAET\_\* (Tipos de entidade de serviços instaláveis)

*Tabela 436. Valores de constantes*

Nome	Valor decimal	Valor Hexadecimal
MQZAET_NONE	0	X'00000000'
MQZAET_PRINCIPAL	1	X'00000001'
MQZAET_GROUP	2	X'00000002'
MQZAET_UNKNOWN	3	X'00000003'

## MQZAO\_\* (Autorizações de Serviços Instaláveis).

*Tabela 437. Valores de constantes*

Nome	Valor decimal	Valor Hexadecimal
MQZAO_CONNECT	1	X'00000001'
MQZAO_BROWSE	2	X'00000002'
MQZAO_INPUT	4	X'00000004'
MQZAO_OUTPUT	8	X'00000008'
MQZAO_INQUIRE	16	X'00000010'
MQZAO_SET	32	X'00000020'
MQZAO_PASS_IDENTITY_CONTEXT	64	X'00000040'
MQZAO_PASS_ALL_CONTEXT	128	X'00000080'
MQZAO_SET_IDENTITY_CONTEXT	256	X'00000100'
MQZAO_SET_ALL_CONTEXT	512	X'00000200'
MQZAO_ALTERNATE_USER_AUTHORITY	1024	X'00000400'
MQZAO_PUBLISH	2048	X'00000800'
MQZAO_SUBSCRIBE	4096	X'00001000'
MQZAO_RESUME	8192	X'00002000'
MQZAO_ALL_MQI	16383	X'00003FFF'
MQZAO_CREATE	65536	X'00010000'
MQZAO_DELETE	131072	X'00020000'
MQZAO_DISPLAY	262144	X'00040000'
MQZAO_CHANGE	524288	X'00080000'
MQZAO_CLEAR	1048576	X'00100000'
MQZAO_CONTROL	2097152	X'00200000'
MQZAO_CONTROL_EXTENDED	4194304	X'00400000'
MQZAO_AUTORIZAR	8388608	X'00800000'
MQZAO_ALL_ADMIN	16646144	X'00FE0000'
MQZAO_ALL	16662527	X'00FE3FFF'
MQZAO_REMOVE	16777216	X'01000000'
MQZAO_NONE	0	X'00000000'

## MQZAS\_\* (Versão da Interface de Serviço de Serviços Instaláveis)

Tabela 438. Valores de constantes		
Nome	Valor decimal	Valor Hexadecimal
MQZAS_VERSION_1	1	X'00000001'
MQZAS_VERSION_2	2	X'00000002'
MQZAS_VERSION_3	3	X'00000003'
MQZAS_VERSION_4	4	X'00000004'
MQZAS_VERSION_5	5	X'00000005'
MQZAS_VERSION_6	6	X'00000006'

## MQZAT\_\* (Tipos de autenticação).

Tabela 439. Valores de constantes		
Nome	Valor decimal	Valor Hexadecimal
MQZAT_INITIAL_CONTEXT	0	X'00000000'
MQZAT_CHANGE_CONTEXT	1	X'00000001'

## MQZCI\_\* (Indicador de continuação de serviços instaláveis)

Tabela 440. Valores de constantes		
Nome	Valor decimal	Valor Hexadecimal
MQZCI_DEFAULT	0	X'00000000'
MQZCI_CONTINUE	0	X'00000000'
MQZCI_STOP	1	X'00000001'

## MQZED\_\* (Estrutura de dados da entidade)

Tabela 441. Estruturas de constantes	
Nome	Estrutura
MQZED_STRUC_ID	"ZED↵"
MQZED_STRUC_ID_ARRAY	'Z','E','D','↵'

**Nota:** O símbolo ↵ representa um único caractere em branco.

Tabela 442. Valores de constantes		
Nome	Valor decimal	Valor Hexadecimal
MQZED_VERSION_1	1	X'00000001'
MQZED_VERSION_2	2	X'00000002'
MQZED_CURRENT_VERSION	2	X'00000002'

## MQZFP\_\* (Estrutura de parâmetros livres)

Tabela 443. Estruturas de constantes	
Nome	Estrutura
MQZFP_STRUC_ID	"ZFP↵"
MQZFP_STRUC_ID_ARRAY	'Z','F','P','↵'

**Nota:** O símbolo ¬ representa um único caractere em branco.

*Tabela 444. Valores de constantes*

Nome	Valor decimal	Valor Hexadecimal
MQZFP_VERSION_1	1	X'00000001'
MQZFP_CURRENT_VERSION	1	X'00000001'

## MQZIC\_\* (Estrutura de contexto de identidade)

*Tabela 445. Estruturas de constantes*

Nome	Estrutura
MQZIC_STRUC_ID	"ZIC¬"
MQZIC_STRUC_ID_ARRAY	'Z','I','C',¬'

**Nota:** O símbolo ¬ representa um único caractere em branco.

*Tabela 446. Valores de constantes*

Nome	Valor decimal	Valor Hexadecimal
MQZIC_VERSION_1	1	X'00000001'
MQZIC_CURRENT_VERSION	1	X'00000001'

## MQZID\_\* (IDs de função para serviços).

### IDs de função comuns a todos os serviços

*Tabela 447. Valores de constantes*

Nome	Valor decimal	Valor Hexadecimal
MQZID_INIT	0	X'00000000'
MQZID_TERM	1	X'00000001'

### IDs de Função para Serviço de Autoridade

*Tabela 448. Valores de constantes*

Nome	Valor decimal	Valor Hexadecimal
MQZID_INIT_AUTHORITY	0	X'00000000'
MQZID_TERM_AUTHORITY	1	X'00000001'
MQZID_CHECK_AUTHORITY	2	X'00000002'
MQZID_COPY_ALL_AUTHORITY	3	X'00000003'
MQZID_DELETE_AUTHORITY	4	X'00000004'
MQZID_SET_AUTHORITY	5	X'00000005'
MQZID_GET_AUTHORITY	6	X'00000006'
MQZID_GET_EXPLICIT_AUTHORITY	7	X'00000007'
MQZID_REFRESH_CACHE	8	X'00000008'
MQZID_ENUMERATE_AUTHORITY_DATA	9	X'00000009'
MQZID_AUTHENTICATE_USER	10	X'0000000A'
MQZID_FREE_USER	11	X'0000000B'
MQZID_INQUIRE	12	X'0000000C'

<i>Tabela 448. Valores de constantes (continuação)</i>		
Nome	Valor decimal	Valor Hexadecimal
MQZID_CHECK_PRIVILEGED	13	X'0000000D'

### IDs de Função para Serviço de Nome

<i>Tabela 449. Valores de constantes</i>		
Nome	Valor decimal	Valor Hexadecimal
MQZID_INIT_NAME	0	X'00000000'
MQZID_TERM_NAME	1	X'00000001'
MQZID_LOOKUP_NAME	2	X'00000002'
MQZID_INSERT_NAME	3	X'00000003'
MQZID_DELETE_NAME	4	X'00000004'

### IDs de função para o serviço Userid

<i>Tabela 450. Valores de constantes</i>		
Nome	Valor decimal	Valor Hexadecimal
MQZID_INIT_USERID	0	X'00000000'
MQZID_TERM_USERID	1	X'00000001'
MQZID_FIND_USERID	2	X'00000002'

### MQZIO\_\* (Opções de inicialização de serviços instaláveis)

<i>Tabela 451. Valores de constantes</i>		
Nome	Valor decimal	Valor Hexadecimal
MQZIO_PRIMARY	0	X'00000000'
MQZIO_SECONDARY	1	X'00000001'

### MQZNS\_\* (Versão da Interface de Serviço de Nomes).

<i>Tabela 452. Valores de constantes</i>		
Nome	Valor decimal	Valor Hexadecimal
MQZNS_VERSION_1	1	X'00000001'

### MQZSE\_\* (Início de Serviços Instaláveis-Indicador de Enumeração)

<i>Tabela 453. Valores de constantes</i>		
Nome	Valor decimal	Valor Hexadecimal
MQZSE_START	1	X'00000001'
MQZSE_CONTINUE	0	X'00000000'

### MQZSL\_\* (Indicador do Seletor de Serviços Instaláveis)

<i>Tabela 454. Valores de constantes</i>		
Nome	Valor decimal	Valor Hexadecimal
MQZSL_NOT_RETORNADO	0	X'00000000'



Tabela 454. Valores de constantes (continuação)		
Nome	Valor decimal	Valor Hexadecimal
MQZSL_RETORNADO	1	X'00000001'

### MQZTO\_\* (Opções de Rescisão de Serviços Instaláveis).

Tabela 455. Valores de constantes		
Nome	Valor decimal	Valor Hexadecimal
MQZTO_PRIMARY	0	X'00000000'
MQZTO_SECONDARY	1	X'00000001'

### MQZUS\_\* (Versão da Interface de Serviço do Usuário)

Tabela 456. Valores de constantes		
Nome	Valor decimal	Valor Hexadecimal
MQZUS_VERSION_1	1	X'00000001'

## Tipos de dados usados no MQI

Informações sobre os tipos de dados que podem ser usados na Message Queue Interface (MQI).  
Descrições, campos e declarações de idioma para idiomas relevantes com cada tipo de dados..

### Tipos de dados e programação para o MQI

Introduzindo os tipos de dados Elementar e de Estrutura e como usar o MQI por meio de programação C, programação COBOL ou programação High Level Assembler .

#### Tipos de dados elementares

Informações sobre os tipos de dados usados no MQI ou nas funções de saída Eles são descritos em detalhes, seguidos por exemplos mostrando como declarar os tipos de dados elementares nas linguagens de programação suportadas.

Os tipos de dados usados no MQI, ou nas funções de saída, são:

- Tipos de dados elementares ou
- Agregados de tipos de dados elementares (matrizes ou estruturas)

Os seguintes tipos de dados elementares são usados no MQI ou nas funções de saída:

Tabela 457. Nomes, tipos e descrições de tipos de dados elementares		
Nome do tipo de dados elementar	Tipo de Dados	Descrição
MQBOOL	Booleana	O tipo de dados MQBOOL representa um valor booleano. O valor 0 representa false. Qualquer outro valor representa true. Um MQBOOL deve estar alinhado como para o tipo de dados MQLONG

Tabela 457. Nomes, tipos e descrições de tipos de dados elementares (continuação)

Nome do tipo de dados elementar	Tipo de Dados	Descrição
MQBYTE	Byte	<p>O tipo de dados MQBYTE representa um único byte de dados. Nenhuma interpretação específica é colocada no byte; ela é tratada como uma sequência de bits, e não como um número binário ou caractere. Não é necessário nenhum alinhamento especial.</p> <p>Quando dados MQBYTE são enviados entre gerenciadores de filas que usam conjuntos de caracteres ou codificações diferentes, os dados MQBYTE não são convertidos de nenhuma maneira. Os campos <i>MsgId</i> e <i>CorrelId</i> na estrutura MQMD são assim.</p> <p>Uma matriz de MQBYTE às vezes é usada para representar uma área de armazenamento principal que não é conhecida pelo gerenciador de filas. Por exemplo, a área pode conter dados de mensagens de aplicativos ou uma estrutura. O alinhamento de limite desta área deve ser compatível com a natureza dos dados contidos dentro dela;</p> <p>Na linguagem de programação C, qualquer tipo de dado pode ser usado para parâmetros de função mostrados como matrizes de MQBYTE. Isso ocorre porque tais parâmetros são sempre transmitidos por endereço, e em C o parâmetro de função é declarado como um ponteiro para vazio.</p>

Tabela 457. Nomes, tipos e descrições de tipos de dados elementares (continuação)

Nome do tipo de dados elementar	Tipo de Dados	Descrição
MQBYTE $n$	Sequência de $n$ bytes	<p>Cada tipo de dados MQBYTE<math>n</math> representa uma sequência de <math>n</math> bytes, em que <math>n</math> pode ter qualquer um dos seguintes valores: 8, 16, 24, 32, 40 ou 128. Cada byte é descrito pelo tipo de dados MQBYTE. Não é necessário nenhum alinhamento especial.</p> <p>Se os dados na sequência de bytes forem menores que o comprimento definido da sequência, os dados deverão ser preenchidos com nulos para preencher a sequência.</p> <p>Quando o gerenciador de filas retorna sequências de bytes para o aplicativo (por exemplo, na chamada MQGET), o gerenciador de filas preenche com nulos o comprimento definido da sequência.</p> <p>As constantes nomeadas estão disponíveis para definir os comprimentos dos campos de sequência de bytes. Eles estão listados em “Constantes” na página <a href="#">61</a>.</p>

Tabela 457. Nomes, tipos e descrições de tipos de dados elementares (continuação)

Nome do tipo de dados elementar	Tipo de Dados	Descrição
MQCHAR	Caractere	<p>O tipo de dados MQCHAR representa um caractere de byte único ou um byte de um caractere de byte duplo ou multibyte. Não é necessário nenhum alinhamento especial.</p> <p>Quando dados MQCHAR são enviados entre gerenciadores de filas que usam conjuntos de caracteres ou codificações diferentes, os dados MQCHAR geralmente requerem conversão para que os dados sejam interpretados corretamente. O gerenciador de filas faz isso automaticamente para dados MQCHAR na estrutura MQMD.. A conversão de dados MQCHAR nos dados da mensagem do aplicativo é controlada pela opção MQGMO_CONVERT especificada na chamada MQGET; consulte a descrição dessa opção em <a href="#">“MQGMO-Obter opções de mensagem” na página 379</a>, para obter detalhes adicionais</p>

Tabela 457. Nomes, tipos e descrições de tipos de dados elementares (continuação)

Nome do tipo de dados elementar	Tipo de Dados	Descrição
MQCHARn	Sequência de $n$ caracteres	<p>Cada tipo de dados MQCHARn representa uma seqüência de <math>n</math> caracteres, em que <math>n</math> pode ter qualquer um dos seguintes valores: 4, 8; 12; 20; 28; 32; 48; 64; 128 ou 256. Cada caractere é descrito por um tipo de dados MQCHAR. Não é necessário nenhum alinhamento especial.</p> <p>Se os dados na seqüência forem menores que o comprimento definido da seqüência, os dados deverão ser preenchidos com espaços em branco para preencher a seqüência. Em alguns casos, um caractere nulo pode ser usado para terminar a seqüência prematuramente, em vez de preenchimento com espaços em branco; o caractere nulo e os caracteres que o seguem são tratados como espaços em branco, até o comprimento definido da seqüência. Os locais em que um nulo pode ser usado são identificados nas descrições de chamada e de tipo de dados</p> <p>Quando o gerenciador de filas retorna seqüências de caracteres para o aplicativo (por exemplo, na chamada MQGET), o gerenciador de filas sempre preenche com espaços em branco para o comprimento definido da seqüência; o gerenciador de filas não usa o caractere nulo para delimitar a seqüência.</p> <p>As constantes nomeadas estão disponíveis que definem os comprimentos dos campos de seqüência de caracteres e são listados em <a href="#">“Constantes” na página 61</a></p>

Tabela 457. Nomes, tipos e descrições de tipos de dados elementares (continuação)

Nome do tipo de dados elementar	Tipo de Dados	Descrição
MQFLOAT32	Número de ponto flutuante de 32 bits	<p>O tipo de dados MQFLOAT32 é um número de vírgula flutuante de 32 bits representado usando o formato de vírgula flutuante IEEE padrão.. Um MQFLOAT32 deve ser alinhado em um limite de 4 bytes</p> <p>O uso de MQFLOAT32 em C on z/OS requer o uso do sinalizador do compilador FLOAT (IEEE).</p> <p>O uso de MQFLOAT32 em COBOL é limitado a compiladores que suportam números de vírgula flutuante em formato IEEE Isso pode requerer o uso do sinalizador do compilador FLOAT (NATIVE)</p>
MQFLOAT64	número de ponto flutuante de 64 bits	<p>O tipo de dados MQFLOAT64 é um número de vírgula flutuante de 64 bits representado usando o formato de vírgula flutuante IEEE padrão Um MQFLOAT64 deve ser alinhado em um limite de 8 bytes</p> <p>O uso de MQFLOAT64 em C no z/OS requer o uso do sinalizador do compilador FLOAT (IEEE).</p> <p>O uso de MQFLOAT64 em COBOL é limitado a compiladores que suportam números de vírgula flutuante no formato IEEE Isso pode requerer o uso do sinalizador do compilador FLOAT (NATIVE)</p>
MQHCONFIG	Identificador de Configuração.	<p>O tipo de dados MQHCONFIG representa um identificador de configuração, ou seja, o componente que está sendo configurado para um serviço instalável específico Uma alça de configuração deve ser alinhada em seu limite natural</p> <p>Os aplicativos não devem contar com o formato dos dados armazenados dentro dessa manipulação Se válido, seu valor deve ser utilizável em chamadas MQI adicionais, mas não deve ter qualquer significado além desse propósito.</p>

Tabela 457. Nomes, tipos e descrições de tipos de dados elementares (continuação)

Nome do tipo de dados elementar	Tipo de Dados	Descrição
MQHCONN	Identificador de conexões	<p>O tipo de dados MQHCONN representa um identificador de conexão, ou seja, a conexão com um gerenciador de filas específico. Um identificador de conexão deve ser alinhado em um limite de 4 bytes.</p> <p>Os aplicativos não devem contar com o formato dos dados armazenados dentro dessa manipulação. Se válido, seu valor deve ser utilizável em chamadas MQI adicionais, mas não deve ter qualquer significado além desse propósito.</p>
MQHMSG	Identificador de mensagens	<p>O tipo de dados MQHMSG representa uma manipulação de mensagem que fornece acesso a uma mensagem. Um identificador de mensagem deve estar alinhado em um limite de 8 bytes.</p> <p>Os aplicativos não devem contar com o formato dos dados armazenados dentro dessa manipulação. Se válido, seu valor deve ser utilizável em chamadas MQI adicionais, mas não deve ter qualquer significado além desse propósito.</p>
MQHOBJ	Manipulação de objetos	<p>O tipo de dados MQHOBJ representa uma manipulação de objetos que fornece acesso a um objeto. Uma manipulação de objetos deve ser alinhada em um limite de 4 bytes..</p> <p>Os aplicativos não devem contar com o formato dos dados armazenados dentro dessa manipulação. Se válido, seu valor deve ser utilizável em chamadas MQI adicionais, mas não deve ter qualquer significado além desse propósito.</p>
MQINT8	Número inteiro assinado de 8 bits	<p>O tipo de dados MQINT8 é um número inteiro assinado de 8 bits que pode usar qualquer valor no intervalo de -128 a +127, a menos que seja restrito de outra forma pelo contexto.</p>

Tabela 457. Nomes, tipos e descrições de tipos de dados elementares (continuação)

Nome do tipo de dados elementar	Tipo de Dados	Descrição
MQINT16	Número inteiro assinado de 16 bits	O tipo de dados MQINT16 é um número inteiro assinado de 16 bits que pode ter qualquer valor no intervalo de -32 768 a +32 767, a menos que seja restrito de outra forma pelo contexto. Um MQINT16 deve ser alinhado em um limite de 2 bytes.
MQINT32	Número inteiro assinado de 32 bits	O tipo de dados MQINT32 é um número inteiro binário assinado de 32 bits que pode ter qualquer valor no intervalo -2 147 483 648 a + 2 147 483 647, a menos que seja restrito de outra forma pelo contexto.  Consulte a definição de <a href="#">MQLONG</a>
MQINT64	Número inteiro assinado de 64 bits	O tipo de dados MQINT64 é um número inteiro assinado de 64 bits que pode usar qualquer valor no intervalo de -9 223 372 036 854 775 808 a + 9 223 372 036 854 775 807, a menos que seja restrito pelo contexto.  Para COBOL, o intervalo válido é limitado a -999 999 999 999 999 999 999 até +999 999 999 999 999 999 999. Um inteiro de 64 bits deve ser alinhado em um limite de 8 bytes.
MQLONG	Número inteiro assinado de 32 bits	O tipo de dados MQLONG é um número inteiro binário assinado de 32 bits que pode ter qualquer valor no intervalo -2 147 483 648 a + 2 147 483 647, a menos que seja restrito de outra forma pelo contexto.  Para COBOL, o intervalo válido é limitado a -999 999 999 a +999 999 999. Um MQLONG deve ser alinhado em um limite de 4 bytes
MQPID	Identificador de Processo	O identificador de processo IBM MQ .  Esse é o mesmo identificador usado no rastreamento do MQ e nos dumps FFST™, mas pode ser diferente do identificador do processo do sistema operacional.



Tabela 457. Nomes, tipos e descrições de tipos de dados elementares (continuação)

Nome do tipo de dados elementar	Tipo de Dados	Descrição
MQPTR	Ponteiro	<p>O tipo de dados MQPTR é o endereço de dados de qualquer tipo.. Um ponteiro deve ser alinhado em seu limite natural; este é um limite de 16 bytes em IBM i, e um limite de 8 bytes em outras plataformas</p> <p>Algumas linguagens de programação suportam ponteiros digitados; o MQI também os usa em alguns casos (por exemplo, PMQCHAR e PMQLONG na linguagem de programação C).</p>
MQTID	Identificador de encadeamento	<p>O identificador de encadeamento do IBM MQ.</p> <p>Esse é o mesmo identificador usado no rastreamento do MQ e FFST™ dumps, mas pode ser diferente do identificador de encadeamento do sistema operacional.</p>
MQUINT8	Número inteiro não assinado de 8 bits	<p>O tipo de dados MQUINT8 é um número inteiro não assinado de 8 bits que pode usar qualquer valor no intervalo de 0 a +255, a menos que seja restrito de outra forma pelo contexto..</p>
MQUINT16	Número inteiro não assinado de 16 bits	<p>O tipo de dados MQUINT16 é um número inteiro não assinado de 16 bits que pode ter qualquer valor no intervalo de 0 a +65 535, a menos que seja restrito de outra forma pelo contexto. Um MQUINT16 deve ser alinhado em um limite de 2 bytes.</p>
MQUINT32	Número inteiro não assinado de 32 bits	<p>O tipo de dados MQUINT32 é um inteiro binário não assinado de 32 bits.</p> <p>Consulte a definição do <a href="#">MQULONG</a>..</p>

Tabela 457. Nomes, tipos e descrições de tipos de dados elementares (continuação)

Nome do tipo de dados elementar	Tipo de Dados	Descrição
MQINT64	Número inteiro não assinado de 64 bits	O tipo de dados MQINT64 é um número inteiro não assinado de 64 bits que pode usar qualquer valor no intervalo de 0 a +18 446 744 073 709 551 615, a menos que seja restrito de outra forma pelo contexto...  Para COBOL, o intervalo válido é limitado a 0 a +999 999 999 999 999 999 999 999. Um inteiro de 64 bits deve ser alinhado em um limite de 8 bytes.
MQULONG	Número inteiro não assinado de 32 bits	O tipo de dados MQULONG é um número inteiro binário não assinado de 32 bits que pode usar qualquer valor no intervalo de 0 a + 4 294 967 294, a menos que seja restrito de outra forma pelo contexto...  Para COBOL, o intervalo válido é limitado a 0 a +999 999 999. Um MQULONG deve ser alinhado em um limite de 4 bytes
PMQACH	Ponteiro	Ponteiro para uma estrutura de dados do tipo MQACH.
PMQAIR	Ponteiro	Ponteiro para uma estrutura de dados do tipo MQAIR.
PMQAXC	Ponteiro	Ponteiro para uma estrutura de dados do tipo MQAXC.
PMQAXP	Ponteiro	Ponteiro para uma estrutura de dados do tipo MQAXP
PMQBMHO	Ponteiro	Ponteiro para uma estrutura de dados do tipo MQBMHO
PMQBO	Ponteiro	Ponteiro para uma estrutura de dados do tipo MQBO
PMQBOOL	Ponteiro	Ponteiro para dados do tipo MQBOOL
PMQBYTE	Ponteiro	Ponteiro para dados do tipo MQBYTE
PMQBYTE <sub>n</sub>	Ponteiro	Ponteiro para dados do tipo MQBYTE <sub>n</sub> , em que n pode ser 8, 16, 24, 32, 40, 128
PMQCBC	Ponteiro	Ponteiro para uma estrutura de dados do tipo MQCBC
PMQCBD	Ponteiro	Ponteiro para uma estrutura de dados do tipo MQCB

Tabela 457. Nomes, tipos e descrições de tipos de dados elementares (continuação)

<b>Nome do tipo de dados elementar</b>	<b>Tipo de Dados</b>	<b>Descrição</b>
PMQCHAR	Ponteiro	Ponteiro para dados do tipo MQCHAR
PMQCHARN	Ponteiro	Ponteiro para um tipo de dados de MQCHARN, em que n pode ser 4, 8, 12, 20, 28, 32, 48, 64, 128, 256, 264
PMQCHARV	Ponteiro	Ponteiro para uma estrutura de dados do tipo MQCHARV..
PMQCIH	Ponteiro	Ponteiro para uma estrutura de dados do tipo MQCIH
PMQCMHO	Ponteiro	Ponteiro para uma estrutura de dados do tipo MQCMHO
PMQCNO	Ponteiro	Ponteiro para uma estrutura de dados do tipo MQCN
PMQCSP	Ponteiro	Ponteiro para uma estrutura de dados do tipo MQCSP
PMQCTLO	Ponteiro	Ponteiro para uma estrutura de dados do tipo MQCTLO
PMQDH	Ponteiro	Ponteiro para uma estrutura de dados do tipo MQDH
PMQDHO	Ponteiro	Ponteiro para uma estrutura de dados do tipo MQDHO
PMQDLH	Ponteiro	Ponteiro para uma estrutura de dados do tipo MQDLH
PMQDMHO	Ponteiro	Ponteiro para uma estrutura de dados do tipo MQDMHO
PMQDMPO	Ponteiro	Ponteiro para uma estrutura de dados do tipo MQDMPO
PMQEPH	Ponteiro	Ponteiro para uma estrutura de dados do tipo MQEPH
PMQFLOAT32	Ponteiro	Ponteiro para uma estrutura de dados do tipo MQFLOAT32
PMQFLOAT64	Ponteiro	Ponteiro para uma estrutura de dados do tipo MQFLOAT64
PMQFUNC	Ponteiro	Ponteiro para uma função
PMQGMO	Ponteiro	Ponteiro para uma estrutura de dados do tipo MQGMO..
PMQHCONFIG	Ponteiro	Ponteiro para dados do tipo MQHCONFIG
PMQHCONN	Ponteiro	Ponteiro para dados do tipo MQHCONN

Tabela 457. Nomes, tipos e descrições de tipos de dados elementares (continuação)

<b>Nome do tipo de dados elementar</b>	<b>Tipo de Dados</b>	<b>Descrição</b>
PMQHMSG	Ponteiro	Ponteiro para dados do tipo MQHMSG
PMQHOBJ	Ponteiro	Ponteiro para dados do tipo MQHOBJ
PMQIIH	Ponteiro	Ponteiro para uma estrutura de dados do tipo MQIIH..
PMQIMPO	Ponteiro	Ponteiro para uma estrutura de dados do tipo MQIMPO
PMQINT8	Ponteiro	Ponteiro para dados do tipo MQINT8
PMQINT16	Ponteiro	Ponteiro para dados do tipo MQINT16
PMQINT32	Ponteiro	Ponteiro para dados do tipo MQINT32
PMQINT64	Ponteiro	Ponteiro para dados do tipo MQINT64
PMQLONG	Ponteiro	Ponteiro para dados do tipo MQLONG
PMQMD	Ponteiro	Ponteiro para estrutura do tipo MQMD
PMQMDE	Ponteiro	Ponteiro para uma estrutura de dados do tipo MQMDE
PMQMD1	Ponteiro	Ponteiro para uma estrutura de dados do tipo MQMD1
PMQMD2	Ponteiro	Ponteiro para uma estrutura de dados do tipo MQMD2
PMQMHBO	Ponteiro	Ponteiro para uma estrutura de dados do tipo MQMHBO
PMQOD	Ponteiro	Ponteiro para uma estrutura de dados do tipo MQOD
PMQOR	Ponteiro	Ponteiro para uma estrutura de dados do tipo MQOR
PMQPD	Ponteiro	Ponteiro para uma estrutura de dados do tipo MQPD
PMQPID	Ponteiro	Ponteiro para um identificador de processo
PMQMD	Ponteiro	Ponteiro para uma estrutura de dados do tipo MQMD..
PMQPMO	Ponteiro	Ponteiro para uma estrutura de dados do tipo MQPMO
PMQPTR	Ponteiro	Ponteiro para dados do tipo MQPTR

Tabela 457. Nomes, tipos e descrições de tipos de dados elementares (continuação)

<b>Nome do tipo de dados elementar</b>	<b>Tipo de Dados</b>	<b>Descrição</b>
PMQRFH	Ponteiro	Ponteiro para uma estrutura de dados do tipo MQRFH
PMQRFH2	Ponteiro	Ponteiro para uma estrutura de dados do tipo MQRFH2
PMQRMH	Ponteiro	Ponteiro para uma estrutura de dados do tipo MQRMH
PMQRR	Ponteiro	Ponteiro para uma estrutura de dados do tipo MQRR
PMQSCO	Ponteiro	Ponteiro para uma estrutura de dados do tipo MQSCO
PMQSD	Ponteiro	Ponteiro para uma estrutura de dados do tipo MQSD
PMQSMPO	Ponteiro	Ponteiro para uma estrutura de dados do tipo MQSMPO
PMQSRO	Ponteiro	Ponteiro para uma estrutura de dados do tipo MQSRO
PMSSTS	Ponteiro	Ponteiro para uma estrutura de dados do tipo MQSTS
PMQTID	Ponteiro	Ponteiro para um ID de encadeamento
PMQTM	Ponteiro	Ponteiro para uma estrutura de dados do tipo MQTM
PMQTM2	Ponteiro	Ponteiro para uma estrutura de dados do tipo MQTM2
PMQUINT8	Ponteiro	Ponteiro para um tipo de dados MQUINT8
PMQUINT16	Ponteiro	Ponteiro para um tipo de dados de MQUINT16
PMQUINT32	Ponteiro	Ponteiro para um tipo de dados MQUINT32
PMQUINT64	Ponteiro	Ponteiro para um tipo de dados MQUINT64
PMQULONG	Ponteiro	Ponteiro para um tipo de dados de MQULONG
PMQVOID	Ponteiro	
PMQWIH	Ponteiro	Ponteiro para uma estrutura de dados do tipo MQWIH
PMQXQH	Ponteiro	Ponteiro para uma estrutura de dados do tipo MQXQH

*Declarações do tipo de dados C*

Tabela 458. C nomes e representações de tipos de dados

<b>Tipo de Dados</b>	<b>Representação</b>
MQBOOL	<code>typedef MQLONG MQBOOL;</code>
MQBYTE	<code>typedef unsigned char MQBYTE;</code>
MQBYTE8	<code>typedef MQBYTE MQBYTE8[8];</code>
MQBYTE16	<code>typedef MQBYTE MQBYTE16[16];</code>
MQBYTE24	<code>typedef MQBYTE MQBYTE24[24];</code>
MQBYTE32	<code>typedef MQBYTE MQBYTE32[32];</code>
MQBYTE40	<code>typedef MQBYTE MQBYTE40[40];</code>
MQCHAR	<code>typedef char MQCHAR;</code>
MQCHAR4	<code>typedef MQCHAR MQCHAR4[4];</code>
MQCHAR8	<code>typedef MQCHAR MQCHAR8[8];</code>
MQCHAR12	<code>typedef MQCHAR MQCHAR12[12];</code>
MQCHAR20	<code>typedef MQCHAR MQCHAR20[20];</code>
MQCHAR28	<code>typedef MQCHAR MQCHAR28[28];</code>
MQCHAR32	<code>typedef MQCHAR MQCHAR32[32];</code>
MQCHAR48	<code>typedef MQCHAR MQCHAR48[48];</code>
MQCHAR64	<code>typedef MQCHAR MQCHAR64[64];</code>

Tabela 458. C nomes e representações de tipos de dados (continuação)

Tipo de Dados	Representação
MQCHAR128	typedef MQCHAR MQCHAR128[128];
MQCHAR256	typedef MQCHAR MQCHAR256[256];
MQFLOAT32	typedef float MQFLOAT32;
MQFLOAT64	typedef double MQFLOAT64;
MQHCONFIG	typedef void MQPOINTER MQHCONFIG;
MQHCONN	typedef MQLONG MQHCONN;
MQHOBJ	typedef MQLONG MQHOBJ;
MQINT8	typedef signed char MQINT8;
MQINT16	typedef short MQINT16;
MQINT64	<p> <span style="background-color: #4F81BD; color: white; padding: 2px;">&gt; UNIX</span> Em UNIX de 64 bits:                      typedef long;                 </p> <p> <span style="background-color: #4F81BD; color: white; padding: 2px;">&gt; AIX</span> Em 32 bits AIX:                      typedef int64_t;                 </p> <p> <span style="background-color: #0070C0; color: white; padding: 2px;">&gt; IBM i</span> <span style="background-color: #4F81BD; color: white; padding: 2px;">&gt; Linux</span> <span style="background-color: #C00000; color: white; padding: 2px;">&gt; z/OS</span> Em Linux, IBM i e z/OS:                      typedef long long;                 </p> <p> <span style="background-color: #C00000; color: white; padding: 2px;">&gt; Windows</span> No Windows:                      typedef _int64;                 </p>

Tabela 458. C nomes e representações de tipos de dados (continuação)










Tipo de Dados	Representação
MQLONG	<p> No IBM i:</p> <pre>typedef long MQLONG;</pre> <p>  Em outras plataformas:</p> <pre>if defined(MQ_64_BIT)     typedef int MQLONG; else     typedef long MQLONG;</pre>
MQPID	<pre>typedef MQLONG MQPID;</pre>
MQPTR	<pre>typedef void MQPOINTER MQPTR;</pre>
MQTID	<pre>typedef MQLONG MQTID;</pre>
MQUINT8	<pre>typedef unsigned char MQUINT8;</pre>
MQUINT16	<pre>typedef unsigned short MQUINT16;</pre>
MQUINT64	<p> Em UNIX de 64 bits:</p> <pre>typedef unsigned long;</pre> <p> Em 32 bits AIX:</p> <pre>typedef uint64_t;</pre> <p>   Em Linux, IBM i e z/OS:</p> <pre>typedef unsigned long long;</pre> <p> No Windows:</p> <pre>typedef unsigned _int64;</pre>



Tabela 458. C nomes e representações de tipos de dados (continuação)




Tipo de Dados	Representação
MQULONG	<p> No IBM i:</p> <pre>typedef unsigned long MQULONG;</pre> <p>  Em outras plataformas:</p> <pre>if defined(MQ_64_BIT)     typedef unsigned int MQULONG; else     typedef unsigned long MQULONG;</pre>
PMQBO	<pre>typedef MQBO MQPOINTER PMQBO;</pre>
PMQBOOL	<pre>typedef MQBOOL MQPOINTER PMQBOOL;</pre>
PMQBYTE	<pre>typedef MQBYTE MQPOINTER PMQBYTE;</pre>
PMQBYTE8	<pre>typedef MQBYTE8[8] MQPOINTER PMQBYTE8[8];</pre>
PMQBYTE16	<pre>typedef MQBYTE16[16] MQPOINTER PMQBYTE16[16];</pre>
PMQBYTE24	<pre>typedef MQBYTE24[24] MQPOINTER PMQBYTE24[24];</pre>
PMQBYTE32	<pre>typedef MQBYTE32[32] MQPOINTER PMQBYTE32[32];</pre>
PMQBYTE40	<pre>typedef MQBYTE40[40] MQPOINTER PMQBYTE40[40];</pre>
PMQBYTE128	<pre>typedef MQBYTE128[128] MQPOINTER PMQBYTE128[128];</pre>
PMQCHAR	<pre>typedef MQCHAR MQPOINTER PMQCHAR;</pre>
PMQCHAR4	<pre>typedef MQCHAR4[4] MQPOINTER PMQCHAR4[4];</pre>
PMQCHAR8	<pre>typedef MQCHAR8[8] MQPOINTER PMQCHAR8[8];</pre>

Tabela 458. C nomes e representações de tipos de dados (continuação)

<b>Tipo de Dados</b>	<b>Representação</b>
PMQCHAR12	<code>typedef MQCHAR12[12] MQPOINTER PMQCHAR12[12];</code>
PMQCHAR20	<code>typedef MQCHAR20[20] MQPOINTER PMQCHAR20[20];</code>
PMQCHAR28	<code>typedef MQCHAR28[28] MQPOINTER PMQCHAR28[28];</code>
PMQCHAR32	<code>typedef MQCHAR32[32] MQPOINTER PMQCHAR32[32];</code>
PMQCHAR48	<code>typedef MQCHAR48[48] MQPOINTER PMQCHAR48[48];</code>
PMQCHAR64	<code>typedef MQCHAR64[64] MQPOINTER PMQCHAR64[64];</code>
PMQCHAR128	<code>typedef MQCHAR128[128] MQPOINTER PMQCHAR128[128];</code>
PMQCHAR256	<code>typedef MQCHAR256[256] MQPOINTER PMQCHAR256[256];</code>
PMQCHAR264	<code>typedef MQCHAR264[264] MQPOINTER PMQCHAR264[264];</code>
PMQCIH	<code>typedef MQCIH MQPOINTER PMQCIH;</code>
PMQCNO	<code>typedef MQCNO MQPOINTER PMQCNO;</code>
PMQDLH	<code>typedef MQDLH MQPOINTER PMQDLH;</code>
PMQFUNC	<code>typedef void MQPOINTER PMQFUNC;</code>
PMQFLOAT32	<code>typedef MQFLOAT32 MQPOINTER PMQFLOAT32;</code>
PMQFLOAT64	<code>typedef MQFLOAT64 MQPOINTER PMQFLOAT64;</code>
PMQGMO	<code>typedef MQGMO MQPOINTER PMQGMO;</code>

Tabela 458. C nomes e representações de tipos de dados (continuação)

<b>Tipo de Dados</b>	<b>Representação</b>
PMQHCONFIG	<code>typedef MQHCONFIG MQPOINTER PMQHCONFIG;</code>
PMQHCONN	<code>typedef MQHCONN MQPOINTER PMQHCONN;</code>
PMQHOBJ	<code>typedef MQHOBJ MQPOINTER PMQHOBJ;</code>
PMQIIH	<code>typedef MQIIH MQPOINTER PMQIIH;</code>
PMQINT8	<code>typedef MQINT8 MQPOINTER PMQINT8;</code>
PMQINT16	<code>typedef MQINT16 MQPOINTER PMQINT16;</code>
PMQLONG	<code>typedef MQLONG MQPOINTER PMQLONG;</code>
PMQMD	<code>typedef MQMD MQPOINTER PMQMD;</code>
PMQMD1	<code>typedef MQMD1[1] MQPOINTER PMQMD1[1];</code>
PMQMDE	<code>typedef MQMDE MQPOINTER PMQMDE;</code>
PMQOD	<code>typedef MQOD MQPOINTER PMQOD;</code>
PMQPMO	<code>typedef MQPMO MQPOINTER PMQPMO;</code>
PMQPTR	<code>typedef MQPTR MQPOINTER PMQPTR;</code>
PMQRFH	<code>typedef MQRFH MQPOINTER PMQRFH;</code>
PMQRFH2	<code>typedef MQRFH2[2] MQPOINTER PMQRFH2[2];</code>
PMQRMH	<code>typedef MQRMH MQPOINTER PMQRMH;</code>

Tabela 458. C nomes e representações de tipos de dados (continuação)

<b>Tipo de Dados</b>	<b>Representação</b>
PMQTM	<code>typedef MQTM MQPOINTER PMQTM;</code>
PMQTM2	<code>typedef MQTM2[2] MQPOINTER PMQTM2[2];</code>
PMQUINT8	<code>typedef MQUINT8 MQPOINTER PMQUINT8;</code>
PMQUINT16	<code>typedef MQUINT16 MQPOINTER PMQUINT16;</code>
PMQULONG	<code>typedef MQULONG MQPOINTER PMQULONG;</code>
PMQVOID	<code>typedef void MQPOINTER PMQVOID;</code>
PMQWIH	<code>typedef MQWIH MQPOINTER PMQWIH;</code>
PMQXQH	<code>typedef MQXQH MQPOINTER PMQXQH;</code>
PPMQBO	<code>typedef PMQBO MQPOINTER PPMQBO;</code>
PPMQBYTE	<code>typedef PMQBYTE MQPOINTER PPMQBYTE;</code>
PPMQCHAR	<code>typedef PMQCHAR MQPOINTER PPMQCHAR;</code>
PPMQCNO	<code>typedef PMQCNO MQPOINTER PPMQCNO;</code>
PPMQGMO	<code>typedef PMQGMO MQPOINTER PPMQGMO;</code>
PPMQHCONN	<code>typedef PMQHCONN MQPOINTER PPMQHCONN;</code>
PPMQHOBJ	<code>typedef PMQHOBJ MQPOINTER PPMQHOBJ;</code>
PPMQLONG	<code>typedef PMQLONG MQPOINTER PPMQLONG;</code>

Tabela 458. C nomes e representações de tipos de dados (continuação)

<b>Tipo de Dados</b>	<b>Representação</b>
PPMQMD	<code>typedef PMQMD MQPOINTER PPMQMD;</code>
PPMQOD	<code>typedef PMQOD MQPOINTER PPMQOD;</code>
PPMQPMO	<code>typedef PMQPMO MQPOINTER PPMQPMO;</code>
PPMQULONG	<code>typedef PMQULONG MQPOINTER PPMQULONG;</code>
PPMQVOID	<code>typedef PMQVOID MQPOINTER PPMQVOID;</code>

em que `defined(MQ_64_BIT)` significa uma plataforma de 64 bits.

Consulte “Tipos de dados” na página 268 para obter uma descrição da variável de macro `MQPOINTER`

*Declarações de tipo de dados COBOL*

Tabela 459. Nomes e representações de tipo de dados COBOL..

<b>Tipo de Dados</b>	<b>Representação</b>
MQBOOL	<code>PIC S9(9) BINARY</code>
MQBYTE	<code>PIC X</code>
MQBYTE8	<code>PIC X(8)</code>
MQBYTE16	<code>PIC X(16)</code>
MQBYTE24	<code>PIC X(24)</code>
MQBYTE32	<code>PIC X(32)</code>
MQBYTE40	<code>PIC X(40)</code>
MQCHAR	<code>PIC X</code>

Tabela 459. Nomes e representações de tipo de dados COBOL.. (continuação)

Tipo de Dados	Representação
MQCHAR4	PIC X(4)
MQCHAR8	PIC X(8)
MQCHAR12	PIC X(12)
MQCHAR20	PIC X(20)
MQCHAR28	PIC X(28)
MQCHAR32	PIC X(32)
MQCHAR48	PIC X(48)
MQCHAR64	PIC X(64)
MQCHAR128	PIC X(128)
MQCHAR256	PIC X(256)
MQFLOAT32	USAGE COMP-1
MQFLOAT64	USAGE COMP-2
MQHCONN	Em z/OS
	PIC S9(9) COMP-5
	Em outras plataformas
	PIC S9(9) BINARY
MQHOBJ	PIC S9(9) BINARY

Tabela 459. Nomes e representações de tipo de dados COBOL.. (continuação)

<b>Tipo de Dados</b>	<b>Representação</b>
MQINT8	PIC S9(2) BINARY
MQINT16	PIC S9(4) BINARY
MQINT64	PIC S9(18) BINARY
MQLONG	PIC S9(9) BINARY
MQPTR	POINTER
MQUINT8	PIC 9(2) BINARY
MQUINT16	PIC 9(4) BINARY
MQUINT64	PIC 9(18) BINARY
MQULONG	PIC 9(9) BINARY

*Declarações de tipo de dados PL/I*

Tabela 460. Nomes e representações de tipo de dados PL/I

<b>Tipo de Dados</b>	<b>Representação</b>
MQBOOL	fixed bin(31)
MQBYTE	char(1)
MQBYTE8	char(8)
MQBYTE16	char(16)
MQBYTE24	char(24)

Tabela 460. Nomes e representações de tipo de dados PL/I (continuação)

<b>Tipo de Dados</b>	<b>Representação</b>
MQBYTE32	char(32)
MQBYTE40	char(40)
MQCHAR	char(1)
MQCHAR4	char(4)
MQCHAR8	char(8)
MQCHAR12	char(12)
MQCHAR20	char(20)
MQCHAR28	char(28)
MQCHAR32	char(32)
MQCHAR48	char(48)
MQCHAR64	char(64)
MQCHAR128	char(128)
MQCHAR256	char(256)
MQFLOAT32	binary float(21) ieee
MQFLOAT64	binary float(52) ieee
MQHCONN	fixed bin(31)



Tabela 460. Nomes e representações de tipo de dados PL/I (continuação)

<b>Tipo de Dados</b>	<b>Representação</b>
MQHOBJ	fixed bin(31)
MQINT8	fixed bin(7)
MQINT16	fixed bin(15)
MQINT64	fixed bin(63)
MQLONG	fixed bin(31)
MQPTR	pointer
MQUINT8	fixed bin(8)
MQUINT16	fixed bin(16)
MQUINT64	fixed bin(64)
MQULONG	fixed bin(32)

*Declarações de tipo de dados High Level Assembler*

Tabela 461. System/390 nomes e representações de tipos de dados do assembler

<b>Tipo de Dados</b>	<b>Representação</b>
MQBOOL	DS F
MQBYTE	DS XL1
MQBYTE8	DS XL8
MQBYTE16	DS XL16

Tabela 461. System/390 nomes e representações de tipos de dados do assembler (continuação)

<b>Tipo de Dados</b>	<b>Representação</b>
MQBYTE24	DS XL24
MQBYTE32	DS XL32
MQBYTE40	DS XL40
MQCHAR	DS CL1
MQCHAR4	DS CL4
MQCHAR8	DS CL8
MQCHAR12	DS CL12
MQCHAR20	DS CL20
MQCHAR28	DS CL28
MQCHAR32	DS CL32
MQCHAR48	DS CL48
MQCHAR64	DS CL64
MQCHAR128	DS CL128
MQCHAR256	DS CL256
MQFLOAT32	DS EB
MQFLOAT64	DS DB

<i>Tabela 461. System/390 nomes e representações de tipos de dados do assembler (continuação)</i>	
<b>Tipo de Dados</b>	<b>Representação</b>
MQHCONN	DS F
MQHOBJ	DS F
MQINT8	DS XL1
MQINT16	DS H
MQINT64	DS D
MQLONG	DS F
MQPTR	DS F
MQUINT8	DS XL1
MQUINT16	DS H
MQUINT64	DS D
MQULONG	DS F

***Tipos de dados da estrutura:***

Um resumo dos tipos de dados de estrutura, regras para mapear as estruturas MQI consistentemente e convenções usadas na descrição de tipo de dados de estrutura.

- [“Resumo dos tipos de dados de estrutura usados em chamadas MQI ou funções de saída” na página 264](#)
- [“Resumo dos tipos de dados da estrutura usados nos dados da mensagem” na página 265](#)
- [“Regras para Mapear as Estruturas MQI Consistentemente” na página 265](#)
- [“Convenções usadas em cada descrição de tipo de dados de estrutura..” na página 266](#)

## Resumo dos tipos de dados de estrutura usados em chamadas MQI ou funções de saída

*Tabela 462. Tipos de dados de estrutura usados em chamadas MQI ou funções de saída*

<b>Estrutura</b>	<b>Descrição</b>	<b>Chamadas onde usadas</b>
MQACH	Cabeçalho da cadeia de saída de API	
<a href="#">MQAIR</a>	Registro de informações sobre autenticação	<a href="#">MQCONN</a>
MQAXC	contexto de saída da API	
MQAXP	parâmetro de saída de API	
<a href="#">MQBMHO</a>	Buffer para opções de manipulação de mensagem	<a href="#">MQBUFMH</a>
<a href="#">MQBO</a>	Iniciar Opções	<a href="#">MQBEGIN</a>
<a href="#">MQCBD</a>	Descritor de retorno de chamada	<a href="#">MQCB</a>
MQCBO	Criar opções de pacote	mqCreateBag
<a href="#">MQCHARV</a>	Sequência de comprimento variável	<a href="#">MQINQMP</a>
<a href="#">MQCNO</a>	Opções de Conexão	<a href="#">MQCONN</a>
<a href="#">MQCSP</a>	Parâmetros de segurança	<a href="#">MQCONN</a>
<a href="#">MQCTLO</a>	Opções de retorno de chamada	<a href="#">MQCTL</a>
<a href="#">MQDMPO</a>	Excluir opções de propriedade de mensagem	<a href="#">MQDLTMP</a>
<a href="#">MQGMO</a>	Opções get-message	<a href="#">MQGet</a>
<a href="#">MQIMPO</a>	Consultar opções de propriedade de mensagem	<a href="#">MQINQMP</a>
<a href="#">MQMD</a>	Descritor de Mensagens	<a href="#">MQBUFMH</a> , <a href="#">MQMHBUF</a> , <a href="#">MQCB</a> , <a href="#">MQGET</a> , <a href="#">MQPUT</a> , <a href="#">MQPUT1</a>
<a href="#">MQMHBO</a>	Identificador de mensagem para opções de buffer.	<a href="#">MQMHBUF</a>
<a href="#">MQOD</a>	Descritor de objeto	<a href="#">MQOPEN</a> , <a href="#">MQPUT1</a>
<a href="#">MQOR</a>	Registro de Objeto	<a href="#">MQOPEN</a> , <a href="#">MQPUT1</a>
<a href="#">MQPD</a>	Descritor de propriedade	<a href="#">MQSETMP</a>
<a href="#">MQPMO</a>	Opções put-message	<a href="#">MQPUT</a> , <a href="#">MQPUT1</a>
<a href="#">MQPMR</a>	Registro de colocar mensagem	<a href="#">MQPUT</a> , <a href="#">MQPUT1</a>
<a href="#">MQRR</a>	Registro de Resposta	<a href="#">MQOPEN</a> , <a href="#">MQPUT</a> , <a href="#">MQPUT1</a>
<a href="#">MQSCO</a>	Opções de configuração TLS	<a href="#">MQCONN</a>
<a href="#">MQSD</a>	Descritor de assinatura	<a href="#">MQSUB</a>
<a href="#">MQSMPO</a>	Configurar opção de propriedade de mensagem	<a href="#">MQSETMP</a>

*Tabela 462. Tipos de dados de estrutura usados em chamadas MQI ou funções de saída (continuação)*

<b>Estrutura</b>	<b>Descrição</b>	<b>Chamadas onde usadas</b>
<u>MQSRO</u>	Opções de solicitação de assinatura	<u>MQSUBRQ</u>
<u>MQSTS</u>	Estrutura de relatório de status	<u>MQSTAT</u>

## Resumo dos tipos de dados da estrutura usados nos dados da mensagem

*Tabela 463. Tipos de dados da estrutura usados nos dados da mensagem*

<b>Estrutura</b>	<b>Descrição</b>
<u>MQCIH</u>	Cabeçalho de informações CICS
<u>MQCFH</u>	Cabeçalho PCF
<u>MQEPH</u>	cabeçalho PCF Integrado
<u>MQDH</u>	Cabeçalho de Distribuição
<u>MQDLH</u>	Cabeçalho de letra morta (mensagem não entregue)
<u>MQIIH</u>	Cabeçalho de informações IMS
<u>MQMDE</u>	Extensão do Descritor de Mensagens
<u>MQRFH</u>	Regras e cabeçalho de formatação
<u>MQRFH2</u>	Regras e cabeçalho de formatação 2
<u>MQRMH</u>	Cabeçalho de Mensagem de Referência
<u>MQTM</u>	Mensagem do acionador
<u>MQTMC2</u>	Mensagem do acionador (formato de caractere 2)
<u>MQWIH</u>	Cabeçalho de Informações de Trabalho
<u>MQXQH</u>	Cabeçalho da fila de transmissão

**Nota:** A estrutura MQDXP (parâmetro de saída de conversão de dados) é descrita em “saída de conversão de dados” na página 939, juntamente com as chamadas de conversão de dados associadas..

## Regras para Mapear as Estruturas MQI Consistentemente

As linguagens de programação variam em seu nível de suporte para estruturas e determinadas regras e convenções são adotadas para mapear as estruturas MQI consistentemente em cada linguagem de programação:

1. As estruturas devem ser alinhadas nos seus limites naturais.
  - A maioria das estruturas de MQI requer alinhamento de 4 bytes
  - No IBM i, estruturas contendo ponteiros requerem alinhamento de 16 bytes; eles são: MQCNO, MQOD, MQPMO.
2. Cada campo em uma estrutura deve ser alinhado em seu limite natural
  - Os campos com tipos de dados que equivalem a MQLONG devem ser alinhados em limites de 4 bytes
  - Campos com tipos de dados que equivalem a MQPTR devem ser alinhados em limites de 16 bytes no IBM i e limites de 4 bytes em outros ambientes.
  - Outros campos são alinhados em limites de 1 byte.

3. O comprimento de uma estrutura deve ser um múltiplo de seu alinhamento de limites
  - A maioria das estruturas MQI tem comprimentos que são múltiplos de 4 bytes
  - No IBM i, as estruturas que contêm ponteiros têm comprimentos que são múltiplos de 16 bytes
4. Quando necessário, os bytes ou campos de preenchimento devem ser incluídos para assegurar a conformidade com as regras anteriores.

## **Convenções usadas em cada descrição de tipo de dados de estrutura..**

A descrição de cada tipo de dados de estrutura inclui:

- Uma visão geral do propósito e uso da estrutura
- Descrições dos campos na estrutura, em um formato que seja independente da linguagem de programação
- Exemplos de como a estrutura é declarada em cada uma das linguagens de programação suportadas

A descrição de cada tipo de dados de estrutura contém as seguintes seções:

### **Nome da Estrutura**

O nome da estrutura, seguido por um resumo dos campos na estrutura..

### **Visão Geral**

Uma breve descrição do propósito e uso da estrutura.

### **Campos**

Descrições dos campos.. Para cada campo, o nome do campo é seguido por seu tipo de dados elementares entre parênteses (). No texto, os nomes de campo são mostrados usando uma fonte itálica; por exemplo, *Version*.

Há também uma descrição do propósito do campo, juntamente com uma lista de quaisquer valores que o campo pode assumir. Nomes de constantes são mostrados em maiúsculas; por exemplo, MQGMO\_STRUC\_ID. Um conjunto de constantes com o mesmo prefixo é mostrado usando o caractere \*, por exemplo: MQIA\_\*.

Nas descrições dos campos, os termos a seguir são usados:

#### **entrada**

Forneça informações no campo ao fazer uma chamada.

#### **saída**

O gerenciador de filas retorna informações no campo quando a chamada é concluída ou falha..

#### **entrada/saída**

Você fornece informações no campo quando faz uma chamada e o gerenciador de filas altera as informações quando a chamada é concluída ou falha.

### **Valores iniciais**

Uma tabela que mostra os valores iniciais para cada campo nos arquivos de definições de dados fornecidos com o MQI

### **Declaração C**

Declaração típica da estrutura em C.

### **Declaração COBOL**

Declaração típica da estrutura em COBOL..

### **Declaração PL/I**

Declaração típica da estrutura em PL/I.

### **Declaração do High Level Assembler**

Declaração típica da estrutura na linguagem assembler do System/390

### **Declaração do Visual Basic**



Declaração típica da estrutura no Visual Basic.

## C programming

Informações para ajudá-lo a usar o MQI da linguagem de programação C.

- [“Arquivos de cabeçalho” na página 267](#)
- [“Funções” na página 267](#)
- [“Parâmetros com o tipo de dados indefinido” na página 268](#)
- [“Tipos de dados” na página 268](#)
- [“Manipulando sequências binárias” na página 268](#)
- [“Manipulação de sequências de caracteres” na página 268](#)
- [“Valores iniciais para estruturas” na página 269](#)
- [“Valores iniciais para as estruturas dinâmicas” na página 269](#)
- [“Uso de C++” na página 270](#)
- [“Convenções de notação” na página 270](#)

### Arquivos de cabeçalho

Tabela 464. Arquivos de cabeçalho em C	
Arquivo	Índice
CMQC	Protótipos de função, tipos de dados e constantes nomeadas para o MQI principal
CMQXC	Protótipos de função, tipos de dados e constantes nomeadas para a saída de conversão de dados
CMQEC	Protótipos de função, tipos de dados e constantes nomeadas para o MQI principal, a saída de conversão de dados e a estrutura de Pontos de Entrada da Interface (CMQEC inclui CMQXC e CMQC.)
CMQSTRC	Funções que convertem definições constantes MQI para o texto equivalente.  <b>Atenção:</b>  Aplicável a z/OS de IBM MQ 9.1. Os programas que utilizam este arquivo de cabeçalho têm de ser compilados com a opção do compilador LONGNAME.

Para melhorar a portabilidade dos aplicativos, codifique o nome do arquivo de cabeçalho em minúsculas na diretriz do pré-processador `#include`:

```
#include "cmqec.h"
```

### Funções

Não é necessário especificar todos os parâmetros que são transmitidos por endereço sempre que você chamar uma função.

- Transmita parâmetros que sejam *somente entrada* e do tipo MQHCONN, MQHOBJ ou MQLONG por valor.
- Transmita todos os outros parâmetros por endereço

Quando um parâmetro específico não for necessário, use um ponteiro nulo como o parâmetro na chamada de função, no lugar do endereço dos dados do parâmetro. Parâmetros para os quais isso é possível estão identificados nas descrições de chamada.

Nenhum parâmetro é retornado como o valor da função; na terminologia C, isso significa que todas as funções retornam `void`.

Os atributos da função são definidos pela variável de macro MQENTRY; o valor dessa variável de macro depende do ambiente.

## Parâmetros com o tipo de dados indefinido

O parâmetro **Buffer** nas funções MQGET, MQPUT e MQPUT1 possui um tipo de dados indefinido.. Esse parâmetro é usado para enviar e receber os dados da mensagem do aplicativo.

Parâmetros desse tipo são mostrados nos exemplos de C como matrizes de MQBYTE. É possível declarar os parâmetros desta maneira, mas geralmente é mais conveniente declará-los como a estrutura específica que descreve o layout dos dados na mensagem Declare o parâmetro de função real como um ponteiro para nulo e especifique o endereço de qualquer tipo de dados como o parâmetro na invocação da função

## Tipos de dados

Defina todos os tipos de dados usando a instrução C typedef Para cada tipo de dados, defina também o tipo de dados do ponteiro correspondente O nome do tipo de dados do ponteiro é o nome do tipo de dados elementar ou de estrutura com o prefixo P para denotar um ponteiro. Defina os atributos do ponteiro usando a variável de macro MQPOINTER; o valor dessa variável de macro depende do ambiente.. O seguinte ilustra como declarar tipos de dados do ponteiro:

```
#define MQPOINTER *                /* depends on environment */
...
typedef MQLONG MQPOINTER PMQLONG; /* pointer to MQLONG */
typedef MQMD MQPOINTER PMQMD;    /* pointer to MQMD */
```

## Manipulando sequências binárias

Declare sequências de dados binários como um dos tipos de dado MQBYTEn.

Sempre que você copiar, comparar ou configurar campos desse tipo, use as funções C **memcpy**, **memcmp** ou **memset** ; por exemplo:

```
#include <string.h>
#include "cmqc.h"

MQMD MyMsgDesc;

memcpy(MyMsgDesc.MsgId,          /* set "MsgId" field to nulls */
       MQMI_NONE,               /* ...using named constant */
       sizeof(MyMsgDesc.MsgId));

memset(MyMsgDesc.CorrelId,       /* set "CorrelId" field to nulls */
       0x00,                    /* ...using a different method */
       sizeof(MQBYTE24));
```

Não use as funções de sequência **strcpy**, **strcmp**, **strncpy** ou **strncmp**, porque elas não funcionam corretamente para dados declarados com os tipos de dados MQBYTEn.

## Manipulação de sequências de caracteres

Quando o gerenciador de filas retornar dados de caracteres para o aplicativo, o gerenciador de filas sempre preencherá os dados de caracteres com espaços em branco para o comprimento definido do campo. O gerenciador de filas *não* retorna sequências nulas.

Portanto, ao copiar, comparar ou concatenar essas sequências, utilize as funções de sequência **strncpy**, **strncmp** ou **strncat**.

Não use as funções de sequência que requerem que a sequência seja finalizada por um nulo (**strcpy**, **strcmp**, **strcat**). Além disso, não use a função **strlen** para determinar o comprimento da sequência; em vez disso, use a função **sizeof** para determinar o comprimento do campo.



## Valores iniciais para estruturas

Os arquivos de cabeçalho definem várias variáveis de macro que podem ser usadas para fornecer valores iniciais para as estruturas MQ ao declarar instâncias dessas estruturas.

Essas variáveis de macro têm nomes no formato MQxxx\_DEFAULT, em que MQxxx representa o nome da estrutura. Eles são usados da seguinte maneira:

```
MQMD    MyMsgDesc = {MQMD_DEFAULT};
MQPMO   MyPutOpts = {MQPMO_DEFAULT};
```

Para alguns campos de caracteres (por exemplo, os campos *StrucId* que ocorrem na maioria das estruturas ou o campo *Format* que ocorre no MQMD), o MQI define valores específicos que são válidos. Para cada um dos valores válidos, duas variáveis de macro são fornecidas:

- Uma variável de macro define o valor como uma cadeia com um comprimento, excluindo as correspondências nulas implícitas, exatamente o comprimento definido do campo. Por exemplo, para o campo *Format* no MQMD, a seguinte variável de macro é fornecida (↵ representa um único caractere em branco):

```
#define MQFMT_STRING "MQSTR↵↵↵"
```

Use este formulário com as funções `memcpy` e `memcmp`

- A outra variável de macro define o valor como uma matriz de caracteres; o nome dessa variável de macro é o nome da forma de sequência sufixada com `_ARRAY`. Por exemplo:

```
#define MQFMT_STRING_ARRAY 'M','Q','S','T','R','↵','↵','↵'
```

Use este formulário para inicializar o campo ao declarar uma instância da estrutura com valores diferentes daqueles fornecidos pela variável de macro `MQMD_DEFAULT`. (Isso nem sempre é necessário; em alguns ambientes, é possível usar a forma de sequência do valor em ambas as situações.) No entanto, você pode usar o formulário de matriz para declarações, porque isso é necessário para compatibilidade com a linguagem de programação C ++.)

## Valores iniciais para as estruturas dinâmicas

Quando um número variável de instâncias de uma estrutura for necessário, as instâncias geralmente serão criadas no armazenamento principal obtido dinamicamente usando as funções `calloc` ou `malloc`. Para inicializar os campos em tais estruturas, considere a seguinte técnica:

1. Declare uma instância da estrutura usando a variável de macro `MQxxx_DEFAULT` apropriada para inicializar a estrutura. Esta instância se torna o modelo para outras instâncias:

```
MQMD Model = {MQMD_DEFAULT}; /* declare model instance */
```

As palavras-chave `static` ou `auto` podem ser codificadas na declaração para fornecer o tempo de vida estático ou dinâmico da instância do modelo, conforme necessário

2. Use as funções `calloc` ou `malloc` para obter armazenamento para uma instância dinâmica da estrutura:

```
PMQMD Instance;
Instance = malloc(sizeof(MQMD)); /* get storage for dynamic instance */
```

3. Use a função `memcpy` para copiar a instância do modelo para a instância dinâmica:

```
memcpy(Instance,&Model,sizeof(MQMD)); /* initialize dynamic instance */
```

## Uso de C++

Para a linguagem de programação C + +, os arquivos de cabeçalho contêm as seguintes instruções adicionais que são incluídas apenas quando você usa um compilador C + +:

```
#ifndef __cplusplus
extern "C" {
#endif

/* rest of header file */

#ifdef __cplusplus
}
#endif
```

## Convenções de notação

Essas informações mostram como chamar as funções e declarar parâmetros.

Em alguns casos, os parâmetros são matrizes com um tamanho não corrigido. Para eles, um n minúsculo é usado para representar uma constante numérica. Ao codificar a declaração para esse parâmetro, substitua n pelo valor numérico necessário.

### Programação COBOL

Informações para ajudá-lo a usar o MQI da linguagem de programação COBOL.

- [“arquivos de cópia” na página 270](#)
- [“Estruturas” na página 271](#)
- [“Ponteiros” na página 272](#)
- [“Constantes nomeadas” na página 272](#)
- [“Convenções de notação” na página 273](#)

## arquivos de cópia

Vários arquivos COPY são fornecidos para ajudar a gravar programas de aplicativos COBOL que usam o MQI. Há dois arquivos contendo constantes nomeadas e dois arquivos para cada uma das estruturas.

Cada estrutura é fornecida em duas formas: um formulário com valores iniciais e um formulário sem:

- Use as estruturas com valores iniciais no WORKING-STORAGE SECTION de um programa COBOL; elas estão contidas em arquivos COPY com nomes sufixados com a letra V (for Values)
- Use as estruturas sem valores iniciais no LINKAGE SECTION de um programa COBOL; elas estão contidas em arquivos COPY com nomes sufixos com a letra L (para Linkage)...

Os arquivos COPY são resumidos na seguinte tabela. Nem todos os arquivos listados estão disponíveis em todos os ambientes..

Arquivo (com valores iniciais)	Arquivo (sem valores iniciais)	Índice
CMQAIRV	CMQAIRL	Registro de informações sobre autenticação
CMQBOV	CMQBOL	Iniciar estrutura de opções
CMQCIHV	CMQCIHL	Estrutura do cabeçalho de informações CICS
CMQCNV	CMQCNOL	Conectar estrutura de opções
CMQDHV	CMQDHL	Estrutura do cabeçalho de distribuição
CMQDLHV	CMQDLHL	Estrutura do cabeçalho de devoluções

Tabela 465. Arquivos de cópia COBOL (continuação)

Arquivo (com valores iniciais)	Arquivo (sem valores iniciais)	Índice
CMQDXPV	CMQDXPL	Estrutura do parâmetro de saída de conversão de dados
CMQGMOV	CMQGMOL	Obter estrutura de opções de mensagem
CMQIIHV	CMQIIHL	Estrutura do cabeçalho de informações IMS
CMQMDV	CMQMDL	Estrutura do descritor de mensagem
CMQMDEV	CMQMDEL	Estrutura de extensão do descritor de mensagens
CMQMD1V	CMQMD1L	Estrutura do descritor de mensagens versão 1
CMQODV	CMQODL	Estrutura do descritor de objeto
CMQORV	CMQORL	Estrutura do registro de objeto
CMQPMOV	CMQPMOL	Estrutura de opções de mensagem put
CMQRFHV	CMQRFHL	Estrutura do cabeçalho de regras e formatação
CMQRFH2V	CMQRFH2L	Regras e estrutura do cabeçalho de formatação versão 2
CMQRMHV	CMQRMHL	Estrutura do cabeçalho da mensagem de referência
CMQRRV	CMQRRL	Estrutura do registro de resposta:
CMQSCOV	CMQSCOL	Opções de configuração TLS
CMQTMV	CMQTML	Estrutura da mensagem do acionador
CMQTMCV	CMQTMCL	Estrutura da mensagem do acionador (formato de caractere).
CMQTM2V	CMQTM2L	Estrutura da mensagem acionadora (formato de caractere) versão 2
CMQWIHV	CMQWIHL	Estrutura do cabeçalho de informações de trabalho
CMQXQHV	CMQXQHL	Estrutura do cabeçalho da fila de transmissão
CMQV	-	Constantes nomeadas para MQI principal
CMQXV	-	Constantes nomeadas para saída de conversão de dados
CMQMD2V	CMQMD2L	Estrutura do descritor de mensagens versão 2

## Estruturas

No arquivo COPY, cada declaração de estrutura começa com um item level-10 ; isso permite declarar várias instâncias da estrutura codificando a declaração level-01 e, em seguida, usando a instrução COPY para copiar o restante da declaração de estrutura. Para consultar a instância apropriada, use a palavra-chave IN :

```
* Declare two instances of MQMD
01 MY-MQMD.
   COPY CMQMDV.
01 MY-OTHER-MQMD.
   COPY CMQMDV.
*
* Set MSGTYPE field in MY-OTHER-MQMD
MOVE MQMT-REQUEST TO MQMD-MSGTYPE IN MY-OTHER-MQMD.
```

Alinhe as estruturas nos limites apropriados Se você usar a instrução COPY para incluir uma estrutura após um item que não seja o item level-01 , assegure-se de que a estrutura comece no deslocamento

apropriado do início do item level-01 . A maioria das estruturas MQI requerem alinhamento de 4 bytes; as exceções para isso são MQCNO, MQOD e MQPMO, que requerem alinhamento de 16 bytes em IBM i

Nesta seção, os nomes de campos em estruturas são mostrados sem um prefixo... Em COBOL, os nomes de campo são prefixados com o nome da estrutura seguido por um hífen. No entanto, se o nome da estrutura terminar com um dígito numérico, indicando que a estrutura é uma segunda versão ou versão posterior da estrutura original, o dígito numérico será omitido do prefixo. Os nomes de campos em COBOL são mostrados em maiúsculas (embora letras minúsculas ou letras maiúsculas e minúsculas possam ser usadas, se necessário). Por exemplo, o campo *MsgType* descrito em [“MQMD - descritor de mensagem” na página 435](#) se torna MQMD-MSGTYPE em COBOL.

As estruturas de sufixo V são declaradas com valores iniciais para todos os campos; é necessário configurar apenas os campos em que você deseja um valor diferente do valor inicial fornecido.

## Ponteiros

Algumas estruturas precisam endereçar dados opcionais que podem ser descontínuos com a estrutura, como os registros MQOR e MQRR endereçados pela estrutura MQOD.

Para endereçar esses dados opcionais, as estruturas contêm campos que são declarados com o tipo de dados do ponteiro. No entanto, o COBOL não suporta o tipo de dados do ponteiro em todos os ambientes. Por isso, os dados opcionais também podem ser endereçados usando campos que contêm o deslocamento dos dados do início da estrutura.

Se desejar transportar um aplicativo entre ambientes, verifique se o tipo de dados do ponteiro está disponível em todos os ambientes desejados. Se não for, o aplicativo deverá endereçar os dados opcionais usando os campos de deslocamento em vez dos campos de ponteiro..

Nos ambientes em que os ponteiros não são suportados, declare os campos de ponteiro como sequências de bytes do comprimento apropriado, com o valor inicial sendo a sequência de bytes all-null. Não altere esse valor inicial se você estiver usando os campos de deslocamento.

## Constantes nomeadas

Nessas informações, os nomes de constantes são mostrados contendo o caractere de sublinhado ( \_ ) como parte do nome. Em COBOL, use o caractere hífen ( - ) no lugar do sublinhado.

Constantes que possuem valores de sequência de caracteres usam aspas simples como o delimitador de sequência ( ' ). Em alguns ambientes, pode ser necessário especificar uma opção do compilador apropriada para fazer com que o compilador aceite aspas simples como o delimitador de sequência no lugar das aspas duplas.

As constantes nomeadas são declaradas nos arquivos COPY como itens de level-10. Para usar as constantes, declare o item level-01 explicitamente e, em seguida, use a instrução COPY para copiar nas declarações de constantes:

```
* Declare a structure to hold the constants
01 MY-MQ-CONSTANTS.
   COPY CMQV.
```

O método anterior faz com que as constantes ocupem o armazenamento no programa mesmo se não forem referenciadas. Se você incluir as constantes em muitos programas separados dentro da mesma unidade de execução, existem várias cópias das constantes, consumindo o armazenamento principal desnecessariamente. Evite este efeito usando uma das técnicas a seguir:

- Inclua a cláusula GLOBAL na declaração level-01 :

```
* Declare a global structure to hold the constants
01 MY-MQ-CONSTANTS GLOBAL.
   COPY CMQV.
```

Isso aloca o armazenamento para apenas um conjunto de constantes dentro da unidade de execução. As constantes, no entanto, podem ser referenciadas por qualquer programa na unidade de execução, não apenas o programa que contém a declaração level-01.

**Nota:** A cláusula GLOBAL não é suportada em todos os ambientes

- Copie manualmente em cada programa apenas as constantes referenciadas por esse programa. Não utilize a instrução COPY para copiar todas as constantes para o programa

## Convenções de notação

As seções anteriores neste tópico mostram como chamar chamadas e declarar parâmetros. Em alguns casos, os parâmetros são tabelas ou sequências de caracteres cujo tamanho não é corrigido. Para eles, um n minúsculo é usado para representar uma constante numérica. Ao codificar a declaração para esse parâmetro, substitua n pelo valor numérico necessário.

## Programação do High Level Assembler

Informações para ajudar a usar o MQI a partir da linguagem de programação Assembler System/390.

- [“Macros” na página 273](#)
- [“Estruturas” na página 274](#)
- [“Macro CMQVERA” na página 274](#)
- [“Convenções de notação” na página 274](#)

## Macros

Há duas macros para constantes nomeadas e uma macro para cada uma das estruturas. Esses arquivos são resumidos na tabela a seguir:

<i>Tabela 466. Macros do assembler.</i>	
<b>Arquivo</b>	<b>Índice</b>
CMQA	Constantes nomeadas (equates) para MQI principal
CMQCIHA	Estrutura do cabeçalho de informações CICS
CMQCNOA	Conectar estrutura de opções
CMQDLHA	Estrutura do cabeçalho de devoluções
CMQDXPA	Estrutura do parâmetro de saída de conversão de dados
CMQGMOA	Obter estrutura de opções de mensagem
CMQIIHA	Estrutura do cabeçalho de informações IMS
CMQMDA	Estrutura do descritor de mensagem
CMQMDEA	Estrutura de extensão do descritor de mensagens
CMQODA	Estrutura do descritor de objeto
CMQPMOA	Estrutura de opções de mensagem put
CMQRFHA	Estrutura do cabeçalho de regras e formatação
CMQRFH2A	Regras e estrutura do cabeçalho de formatação versão 2
CMQRMHA	Estrutura do cabeçalho da mensagem de referência
CMQTMA	Estrutura da mensagem do acionador
CMQTMC2A	Estrutura da mensagem acionadora (formato de caractere) versão 2
CMQVERA	Controle de versão da estrutura

Tabela 466. Macros do assembler. (continuação)

Arquivo	Índice
CMQWIHA	Estrutura do cabeçalho de informações de trabalho
CMQXA	Constantes nomeadas para saída de conversão de dados
CMQXPA	Estrutura do parâmetro de saída cruzada da API
CMQXQHA	Estrutura do cabeçalho da fila de transmissão

## Estruturas

As estruturas são geradas por macros que possuem vários parâmetros para controlar a ação da macro. Consulte a [“Estruturas” na página 274](#)

## Macro CMQVERA

Essa macro permite configurar o valor padrão a ser usado para o parâmetro DCLVER nas macros de estrutura.

O valor especificado por CMQVERA será usado pela macro de estrutura somente se você omitir o parâmetro DCLVER da chamada da macro de estrutura... O valor padrão é configurado codificando a macro CMQVERA com o parâmetro DCLVER :

### DCLVER=CURRENT

A versão padrão é configurada para a versão atual (mais recente)..

### DCLVER=ESPECIFICADO

A versão padrão é configurada para a versão especificada no parâmetro VERSION ..

Deve-se especificar o parâmetro **DCLVER** e o valor deve estar em maiúsculas. O valor configurado por CMQVERA permanece o valor padrão até a próxima chamada de CMQVERA ou o final do conjunto. Se você omitir CMQVERA, o padrão será DCLVER=CURRENT.

## Convenções de notação

Outros tópicos mostram como chamar as chamadas e declarar parâmetros. Em alguns casos, os parâmetros são matrizes ou sequências de caracteres com um tamanho que não é fixo para o qual um *n* minúsculo é usado para representar uma constante numérica. Ao codificar a declaração para esse parâmetro, substitua *n* pelo valor numérico necessário.

### Estruturas

As estruturas são geradas por macros que possuem vários parâmetros para controlar a ação da macro.

**Nota:** De tempos em tempos novas versões das estruturas IBM MQ são introduzidas. Os campos adicionais em uma nova versão podem fazer com que uma estrutura que anteriormente era menor que 256 bytes se torne maior que 256 bytes. Devido a isso, grave instruções do assembler destinadas a copiar uma estrutura IBM MQ ou a configurar uma estrutura IBM MQ para nulos, para trabalhar corretamente com estruturas que podem ser maiores que 256 bytes. Como alternativa, use o parâmetro da macro DCLVER ou a macro CMQVERA com o parâmetro VERSION para declarar uma versão específica da estrutura.

- [“Especificando o nome da estrutura” na página 275](#)
- [“Especificando a forma da estrutura” na página 275](#)
- [“Controlando a versão da estrutura” na página 275](#)
- [“Declarando uma estrutura integrada em outra” na página 275](#)
- [“Especificando valores iniciais para campos” na página 276](#)
- [“Controlando a listagem” na página 276](#)

## Especificando o nome da estrutura

Para declarar mais de uma instância de uma estrutura, a macro prefixa o nome de cada campo na estrutura com uma sequência especificada pelo usuário e um sublinhado.

A cadeia utilizada é o rótulo especificado na chamada da macro.. Se nenhum rótulo for especificado, o nome da estrutura será usado para construir o prefixo:

```
* Declare two object descriptors
      CMQODA ,          Prefix used="MQOD_" (the default)
MY_MQOD CMQODA ,      Prefix used="MY_MQOD_"
```

As declarações de estrutura mostradas nesta seção usam o prefixo padrão

## Especificando a forma da estrutura

As declarações de estrutura podem ser geradas pela macro em uma de duas formas, controladas pelo parâmetro DSECT :

### DSECT=YES

Uma instrução DSECT do assembler é usada para iniciar uma nova seção de dados; a definição de estrutura segue imediatamente a instrução DSECT O rótulo na chamada da macro é usado como o nome da seção de dados; se nenhum rótulo for especificado, o nome da estrutura será usado.

### DSECT=NO

As instruções DC do assembler são usadas para definir a estrutura na posição atual na rotina Os campos são inicializados com valores, que podem ser especificados codificando os parâmetros relevantes na chamada da macro. Campos para os quais nenhum valor é especificado na chamada de macro são inicializados com valores padrão.

O valor especificado deve estar em maiúsculas. Se o parâmetro DSECT não for especificado, DSECT = NO será assumido.

## Controlando a versão da estrutura

Por padrão, as macros sempre declaram a versão mais recente de cada estrutura..

Embora seja possível usar o parâmetro da macro VERSION para especificar um valor para o campo *Version* na estrutura, esse parâmetro define o valor inicial do campo *Version* e não controla a versão da estrutura realmente declarada. Para controlar a versão da estrutura que é declarada, use o parâmetro DCLVER :

### DCLVER=CURRENT

A versão declarada é a atual (mais recente) versão.

### DCLVER=ESPECIFICADO

A versão declarada é a versão especificada pelo parâmetro VERSION Se você omitir o parâmetro VERSION , o padrão será versão 1.

Se você especificar o parâmetro VERSION , o valor deverá ser uma constante numérica de autodefinição ou a constante nomeada para a versão necessária (por exemplo MQCNO\_VERSION\_3) Se você especificar algum outro valor, a estrutura será declarada como se DCLVER=CURRENT tivesse sido especificada, mesmo se o valor de VERSION for resolvido para um valor válido..

O valor especificado deve estar em maiúsculas. Se você omitir o parâmetro DCLVER , o valor usado será obtido da variável de macro global MQDCLVER . É possível configurar essa variável usando a macro CMQVERA.

## Declarando uma estrutura integrada em outra

Para declarar uma estrutura como um componente de outra estrutura, use o parâmetro NESTED :

### NESTED=YES

A declaração de estrutura é aninhada em outra.

## NESTED=NO

A declaração de estrutura não está aninhada em outra.

O valor especificado deve estar em maiúsculas. Se você omitir o parâmetro NESTED , NESTED=NO será assumido.

## Especificando valores iniciais para campos

Especifique o valor a ser utilizado para inicializar um campo em uma estrutura, codificando o nome desse campo (sem o prefixo) como um parâmetro na chamada de macro, acompanhado pelo valor necessário...

Por exemplo, para declarar uma estrutura de descritor de mensagem com o campo *MsgType* inicializado com MQMT\_REQUEST e o campo *ReplyToQ* inicializado com a sequência "MY\_REPLY\_TO\_QUEUE", use o seguinte:

```
MY_MQMD  CMQMDA  MSGTYPE=MQMT_REQUEST,          X
          REPLYTOQ=MY_REPLY_TO_QUEUE
```

Se você especificar uma constante nomeada (equate) como um valor na chamada da macro, use a macro CMQA para definir a constante nomeada. Não coloque os valores da cadeia de caracteres entre aspas simples

## Controlando a listagem

Controle a aparência da declaração de estrutura na listagem do assembler usando o parâmetro LIST :

### LIST=YES

A declaração de estrutura aparece na listagem do assembler

### LIST=NO

A declaração de estrutura não aparece na listagem do assembler

O valor especificado deve estar em maiúsculas. Se você omitir o parâmetro LIST , LIST = NO será assumido.

## MQAIR-Registro de informações sobre autenticação.

A estrutura MQAIR permite que um aplicativo em execução como um IBM MQ MQI client especifique informações sobre um autenticador que será usado para a conexão do cliente. A estrutura é um parâmetro de saída na chamada MQCONN.

## Disponibilidade

A estrutura MQAIR está disponível para os clientes a seguir:

-  AIX
-  Linux
-  Windows

## Conjunto de caracteres e codificação

Os dados no MQAIR devem estar no conjunto de caracteres e na codificação do gerenciador de fila local; eles são fornecidos pelo atributo do gerenciador de filas do **CodedCharSetId** e MQENC\_NATIVE

## Campos

**Nota:** Na tabela a seguir, os campos são agrupados por uso e não alfabeticamente. Os tópicos filhos seguem a mesma sequência.



Tabela 467. Campos em MQAIR

Nome e descrição do campo	Nome da constante	Valor inicial (se houver) da constante
StrucId (identificador de estrutura)	MQAIR_STRUC_ID	'AIR↵'
Versão (número da versão da estrutura).	MQAIR_VERSION_1	1
AuthInfoTipo (tipo de informações sobre autenticação)	MQAIT_CRL_LDAP	1
AuthInfoConnName (nome da conexão do servidor LDAP CRL)	Nenhum	Cadeia nula ou espaços em branco
LDAPUserNamePtr (endereço do nome do usuário LDAP)	Nenhum	Ponteiro nulo ou bytes nulos
LDAPUserNameDeslocamento (deslocamento do nome do usuário LDAP do início de MQSCO)	Nenhum	0
LDAPUserNameComprimento (comprimento do nome do usuário LDAP)	Nenhum	0
LDAPPassword (senha para acessar o servidor LDAP)	Nenhum	Cadeia nula ou espaços em branco
<b>Nota:</b> Os campos restantes serão ignorados se a <i>Versão</i> for menor que MQAIR_VERSION_2.		
OCSPResponderURL (URL na qual o respondente OCSP pode ser contatado)	Nenhum	Cadeia nula ou espaços em branco
<p><b>Notas:</b></p> <ol style="list-style-type: none"> <li>1. O símbolo ↵ representa um único caractere em branco.</li> <li>2. Na linguagem de programação C, a variável macro MQAIR_DEFAULT contém os valores listados na tabela. Use-o da seguinte maneira para fornecer valores iniciais para os campos na estrutura:</li> </ol> <pre style="background-color: #f0f0f0; padding: 10px;">MQAIR MyAIR = {MQAIR_DEFAULT};</pre>		

## Declarações de idiomas

### Declaração C para MQAIR

```
typedef struct tagMQAIR MQAIR;
struct tagMQAIR {
    MCHAR4    StrucId;           /* Structure identifier */
    MQLONG    Version;          /* Structure version number */
    MQLONG    AuthInfoType;     /* Type of authentication
                                information */
    MCHAR264  AuthInfoConnName; /* Connection name of CRL LDAP
                                server */
    PMCHAR    LDAPUserNamePtr;  /* Address of LDAP user name */
    MQLONG    LDAPUserNameOffset; /* Offset of LDAP user name from start
                                of MQAIR structure */
    MQLONG    LDAPUserNameLength; /* Length of LDAP user name */
    MCHAR32   LDAPPassword;     /* Password to access LDAP server */
    MCHAR256  OCSPResponderURL; /* URL of OCSP responder */
};
```

### Declaração COBOL para MQAIR

```
** MQAIR structure
```

```

10 MQAIR.
**  Structure identifier
15  MQAIR-STRUCID          PIC X(4).
**  Structure version number
15  MQAIR-VERSION        PIC S9(9) BINARY.
**  Type of authentication information
15  MQAIR-AUTHINFOTYPE    PIC S9(9) BINARY.
**  Connection name of CRL LDAP server
15  MQAIR-AUTHINFOCONNNAME PIC X(264).
**  Address of LDAP user name
15  MQAIR-LDAPUSERNAMEPTR  POINTER.
**  Offset of LDAP user name from start of MQAIR structure
15  MQAIR-LDAPUSERNAMEOFFSET PIC S9(9) BINARY.
**  Length of LDAP user name
15  MQAIR-LDAPUSERNAMELENGTH PIC S9(9) BINARY.
**  Password to access LDAP server
15  MQAIR-LDAPPASSWORD     PIC X(32).
**  URL of OCSP responder
15  MQAIR-OCSPRESPONDERURL PIC X(256).

```

## Declaração Visual Basic para MQAIR

```

Type MQAIR
  StrucId      As String*4  'Structure identifier'
  Version      As Long      'Structure version number'
  AuthInfoType As Long      'Type of authentication information'
  AuthInfoConnName As String*264 'Connection name of CRL LDAP server'
  LDAPUserNamePtr As MQPTR   'Address of LDAP user name'
  LDAPUserNameOffset As Long  'Offset of LDAP user name from start'
                                'of MQAIR structure'
  LDAPUserNameLength As Long  'Length of LDAP user name'
  LDAPPASSWORD As String*32  'Password to access LDAP server'
End Type

```

### **StrucId (MQCHAR4) para MQAIR**

Este é o identificador da estrutura do registro de informações sobre autenticação. Ele é sempre um campo de entrada Seu valor é MQAIR\_STRUC\_ID.

O valor deve ser:.

#### **MQAIR\_STRUC\_ID**

Identificador para o registro de informações sobre autenticação.

Para a linguagem de programação C, a constante MQAIR\_STRUC\_ID\_ARRAY também é definida. Ele tem o mesmo valor que MQAIR\_STRUC\_ID, mas é uma matriz de caracteres em vez de uma cadeia.

### **Versão (MQLONG) para MQAIR**

Este é o número da versão da estrutura do registro das informações sobre autenticação Ele é sempre um campo de entrada

O valor deve ser um dos seguintes:

#### **MQAIR\_VERSION\_1**

Registro de informações sobre autenticação Version-1 .

Esse é o valor inicial desse campo.

#### **MQAIR\_VERSION\_2**

Registro de informações sobre autenticação Version-2 .

A constante a seguir especifica o número da versão atual:

#### **MQAIR\_CURRENT\_VERSION**

A versão atual do registro de informações sobre autenticação.

### **AuthInfoTipo (MQLONG) para MQAIR**

Este é o tipo de informações sobre autenticação contidas no registro

O valor pode ser um dos dois parâmetros a seguir:

## **MQAIT\_CRL\_LDAP**

Verificação de revogação de certificado usando o servidor LDAP

## **MQAIT\_OCSP**

Verificação de revogação de certificado usando o OCSP

Se o valor não for válido, a chamada falhará com código de razão MQRC\_AUTH\_INFO\_TYPE\_ERROR.

Este é um campo de entrada. O valor inicial desse campo é MQAIT\_CRL\_LDAP.

## **AuthInfoConnName (MQCHAR264) para MQAIR**

Este é o nome do host ou o endereço de rede de um host no qual o servidor LDAP está em execução

Este pode ser seguido por um número de porta opcional, colocado entre parênteses. O número da porta padrão é 389.

Se o valor for menor que o comprimento do campo, termine o valor com um caractere nulo ou preencha-o com espaços em branco até o comprimento do campo. Se o valor não for válido, a chamada falhará com código de razão MQRC\_AUTH\_INFO\_CONN\_NAME\_ERROR.

Este é um campo de entrada. O comprimento deste campo é fornecido por MQ\_AUTH\_INFO\_CONN\_NAME\_LENGTH. O valor inicial desse campo é a cadeia nula em C e caracteres em branco em outras linguagens de programação.

## **LDAPUserNamePtr (PMQCHAR) para MQAIR**

Este é o nome do usuário do LDAP

Ele consiste no Nome Distinto do usuário que está tentando acessar o servidor CRL do LDAP. Se o valor for menor que o comprimento especificado por *LDAPUserNameLength*, finalize o valor com um caractere nulo ou preencha com espaços em branco para o comprimento *LDAPUserNameLength*. O campo será ignorado se *LDAPUserNameLength* for zero.

É possível fornecer o nome do usuário LDAP de uma das duas maneiras:

- Usando o campo de ponteiro *LDAPUserNamePtr*.

Nesse caso, o aplicativo pode declarar uma sequência separada da estrutura MQAIR e configurar *LDAPUserNamePtr* para o endereço da sequência.

Considere o uso do *LDAPUserNamePtr* para linguagens de programação que suportam o tipo de dados do ponteiro de uma maneira que seja móvel para ambientes diferentes (por exemplo, a linguagem de programação C)

- Usando o campo de deslocamento *LDAPUserNameOffset*

Nesse caso, o aplicativo deve declarar uma estrutura composta contendo a estrutura MQSCO seguida pela matriz de registros MQAIR seguidos pelas sequências de nome do usuário LDAP e configurar *LDAPUserNameOffset* para o deslocamento da sequência de nome apropriada do início da estrutura MQAIR. Assegure-se de que esse valor esteja correto e tenha um valor que possa ser acomodado em um MQLONG (a linguagem de programação mais restritiva é COBOL, para o qual o intervalo válido é -999 999 999 a +999 999 999).

Considere o uso do *LDAPUserNameOffset* para linguagens de programação que não suportam o tipo de dados do ponteiro ou que implementam o tipo de dados do ponteiro de uma maneira que pode não ser móvel para ambientes diferentes (por exemplo, a linguagem de programação COBOL)..

Qualquer técnica escolhida, use apenas um de *LDAPUserNamePtr* e *LDAPUserNameOffset*; a chamada falha com o código de razão MQRC\_LDAP\_USER\_NAME\_ERROR se ambos forem diferentes de zero.

Este é um campo de entrada. O valor inicial deste campo é o ponteiro nulo nas linguagens de programação que suportam ponteiros e, caso contrário, uma cadeia de bytes all-null.

**Nota:** Nas plataformas em que a linguagem de programação não suporta o tipo de dados do ponteiro, este campo é declarado como uma cadeia de bytes do comprimento apropriado.

### ***LDAPUserNameDeslocamento (MQLONG) para MQAIR***

Este é o deslocamento em bytes do nome do usuário LDAP do início da estrutura MQAIR.

O deslocamento pode ser positivo ou negativo. O campo será ignorado se *LDAPUserNameLength* for zero.

É possível usar um *LDAPUserNamePtr* ou *LDAPUserNameOffset* para especificar o nome do usuário LDAP, mas não ambos; consulte a descrição do campo *LDAPUserNamePtr* para obter detalhes.

Este é um campo de entrada. O valor inicial desse campo é 0.

### ***LDAPUserNameComprimento (MQLONG) para MQAIR***

Esse é o comprimento em bytes do nome do usuário LDAP endereçado pelo campo *LDAPUserNamePtr* ou *LDAPUserNameOffset*.

O valor deve estar no intervalo zero até *MQ\_DISTINGUISHED\_NAME\_LENGTH*. Se o valor não for válido, a chamada falha com o código de razão *MQRC\_LDAP\_USER\_NAME\_LENGTH\_ERR*.

Se o servidor LDAP envolvido não requerer um nome de usuário, configure esse campo como zero.

Este é um campo de entrada. O valor inicial desse campo é 0.

### ***LDAPPassword (MQCHAR32) para MQAIR***

Essa é a senha necessária para acessar o servidor de CRL LDAP. Se o valor for menor que o comprimento do campo, termine o valor com um caractere nulo ou preencha-o com espaços em branco até o comprimento do campo.

Se o servidor LDAP não requerer uma senha ou você omitir o nome do usuário LDAP, *LDAPPassword* deverá ser nulo ou em branco. Se você omitir o nome do usuário LDAP e *LDAPPassword* não for nulo ou em branco, a chamada falhará com o código de razão *MQRC\_LDAP\_PASSWORD\_ERROR*.

Este é um campo de entrada. O comprimento desse campo é fornecido por *MQ\_LDAP\_PASSWORD\_LENGTH*. O valor inicial desse campo é a cadeia nula em C e caracteres em branco em outras linguagens de programação.

### ***OCSPResponderURL (MQCHAR256) para MQAIR.***

Para uma estrutura MQAIR que representa detalhes da conexão para um respondente do OCSP, esse campo contém a URL na qual o respondente pode ser contatada.

O valor desse campo é uma URL HTTP URL.. Esse campo tem prioridade sobre uma URL em uma extensão de certificado de AuthorityInfoAccess (AIA).

O valor é ignorado, a menos que ambas as seguintes instruções sejam verdadeiras:

- A estrutura MQAIR é Versão 2 ou posterior (o campo Versão é configurado como *MQAIR\_VERSION\_2* ou superior).
- O campo Tipo AuthInfo é configurado como *MQAIT\_OCSP*.

Se o campo não contiver uma URL HTTP no formato correto (e não estiver sendo ignorada), a chamada *MQCONN* falhará com o código de razão *MQRC\_OCSP\_URL\_ERROR*.

Esse campo faz distinção entre maiúsculas e minúsculas. Ele deve iniciar com a cadeia *http://* em minúsculas. O restante da URL pode fazer distinção entre maiúsculas e minúsculas, dependendo da implementação do servidor OCSP.

Este campo não está sujeito à conversão de dados.

### ***MQBMHO-Opções de manipulação de buffer para mensagem***

A estrutura MQBMHO permite que os aplicativos especifiquem opções que controlam como as manipulações de mensagens são produzidas a partir de buffers. A estrutura é um parâmetro de entrada na chamada de *MQBUF*.

## Conjunto de caracteres e codificação

Os dados em MQBMHO devem estar no conjunto de caracteres do aplicativo e na codificação do aplicativo (MQENC\_NATIVE)...

## Campos

**Nota:** Na tabela a seguir, os campos são agrupados por uso e não alfabeticamente. Os tópicos filhos seguem a mesma sequência.

Tabela 468. Campos em MQBMHO		
Nome e descrição do campo	Nome da constante	Valor inicial (se houver) da constante
StrucId (identificador de estrutura)	MQBMHO_STRUC_ID	'BMHO'
Versão (número da versão da estrutura).	MQBMHO_VERSION_1	1
Opções (opções que controlam a ação MQBMHO)	MQBMHO_NONE	0

**Notas:**

1. Na linguagem de programação C, a variável macroMQBMHO\_DEFAULT contém os valores listados na tabela. Use-o da seguinte maneira para fornecer valores iniciais para os campos na estrutura:

```
MQBMHO MyBMHO = {MQBMHO_DEFAULT};
```

## Declarações de idiomas

Declaração C para MQBMHO

```
typedef struct tagMQBMHO MQBMHO;
struct tagMQBMHO {
    MQCHAR4  StructId;          /* Structure identifier */
    MQLONG   Version;          /* Structure version number */
    MQLONG   Options;          /* Options that control the action of
                               MQBUFMH */
};
```

Declaração COBOL para MQBMHO

```
** MQBMHO structure
10 MQBMHO.
** Structure identifier
15 MQBMHO-STRUCID          PIC X(4).
** Structure version number
15 MQBMHO-VERSION        PIC S9(9) BINARY.
** Options that control the action of MQBUFMH
15 MQBMHO-OPTIONS        PIC S9(9) BINARY.
```

Declaração PL/I para MQBMHO

```
Dcl
  1 MQBMHO based,
  3 StructId      char(4),          /* Structure identifier */
  3 Version       fixed bin(31), /* Structure version number */
  3 Options       fixed bin(31), /* Options that control the action
                               of MQBUFMH */
```

Declaração High Level Assembler para MQBMHO

```
MQBMHO          DSECT
```

MQBMHO_STRUCID	DS	CL4	Structure identifier
MQBMHO_VERSION	DS	F	Structure version number
MQBMHO_OPTIONS	DS	F	Options that control the action of MQBUFMH
*			
MQBMHO_LENGTH	EQU	*-MQBMHO	
MQBMHO_AREA	DS	CL(MQBMHO_LENGTH)	

### **StrucId (MQCHAR4) para MQBMHO para MQBMHO.**

Este é o identificador de estrutura do buffer para estrutura de manipulação de mensagens. Ele é sempre um campo de entrada Seu valor é MQBMHO\_STRUC\_ID..

O valor deve ser:.

#### **MQBMHO\_STRUC\_ID**

Identificador para a estrutura de manipulação de buffer para mensagem.

Para a linguagem de programação C, a constante MQBMHO\_STRUC\_ID\_ARRAY também é definida. Ele tem o mesmo valor que MQBMHO\_STRUC\_ID, mas é uma matriz de caracteres em vez de uma cadeia.

### **Versão (MQLONG) para MQBMHO para MQBMHO**

Este é o número da versão do buffer para a estrutura de manipulação de mensagens Ele é sempre um campo de entrada

O valor deve ser:.

#### **MQBMHO\_VERSION\_1**

Número da versão para o buffer para a estrutura de manipulação de mensagens

A constante a seguir especifica o número da versão atual:

#### **MQBMHO\_CURRENT\_VERSION**

Versão atual da estrutura de buffer para manipulação de mensagens.

### **Opções (MQLONG) para MQBMHO**

Estrutura do buffer para manipulação de mensagem-campo Opções

O valor do pode ser:

#### **MQBMHO\_DELETE\_PROPERTIES**

As propriedades incluídas na manipulação de mensagens são excluídas do buffer. Se a chamada falhar nenhuma propriedade será excluída.

Opções padrão: Se você não precisar da opção descrita, use a seguinte opção:

#### **MQBMHO\_NONE**

Nenhuma opção especificada.

Isto é sempre um campo de entrada. O valor inicial desse campo é MQBMHO\_DELETE\_PROPERTIES.

## **V 9.3.0 MQBNO-Opções de balanceamento**

A tabela a seguir resume os campos na estrutura

### **Campos**

**Nota:** Na tabela a seguir, os campos são agrupados por uso e não alfabeticamente. Os tópicos filhos seguem a mesma sequência.

<i>Tabela 469. Campos em MQBNO</i>		
<b>Nome e descrição do campo</b>	<b>Nome da constante</b>	<b>Valor inicial (se houver) da constante</b>
StrucId (identificador de estrutura)	MQBNO_STRUC_ID	'BNO↵'

Tabela 469. Campos em MQBNO (continuação)

Nome e descrição do campo	Nome da constante	Valor inicial (se houver) da constante
Versão (número da versão da estrutura).	MQBNO_VERSION_1	1
ApplicationType (tipo de opção de balanceamento configurado na estrutura)	MQBNO_VALTYPE_SIMPLE	0
Tempo Limite (tempo limite após o qual o rebalanceamento pode interromper a atividade do aplicativo)	MQBNO_TIMEOUT_AS_DEFAULT	0
BalanceOptions (opções de balanceamento configuradas pelo aplicativo emissor)	MQBNO_OPTIONS_NONE	0

**Notas:**

1. O símbolo – representa um único caractere em branco.
2. Na linguagem de programação C, a variável de macro MQBNO\_DEFAULT contém os valores listados na tabela. Use-o da seguinte maneira para fornecer valores iniciais para os campos na estrutura:

```
MQBNO MyBNO = {MQBNO_DEFAULT};
```

## Declarações de idiomas

### Declaração C para MQBNO

```
typedef struct tagMQBNO MQBNO;
struct tagMQBNO {
    MQCHAR4    StrucId;           /* Structure identifier */
    MQLONG     Version;          /* Structure version number */
    MQLONG     Type;             /* Type of balancing options set in the
                                structure */
    MQLONG     Timeout;          /* Timeout after which re-balancing might
                                interrupt application activity */
    MQLONG     BalanceOptions;   /* Balancing options set by the issuing
                                application */
};
```

### Declaração COBOL para MQBNO

```
** MQBNO structure
10 MQBNO.
** Structure identifier
15 MQBNO-STRUCID PIC X(4).
** Structure version number
15 MQBNO-VERSION PIC S9(9) BINARY.
** Type of balancing options set in the structure
15 MQBNO-TYPE PIC S9(9) BINARY.
** Timeout after which re-balancing might interrupt application activity
15 MQBNO-TIMEOUT PIC S9(9) BINARY.
** Balancing options set by the issuing application
15 MQBNO-BALANCEOPTIONS PIC S9(9) BINARY.
```

### Declaração PL/I para MQBNO

```
dcl
1 MQBNO based,
3 StrucId char(4), /* Structure identifier */
3 Version fixed bin(31), /* Structure version number */
3 Type fixed bin(31), /* Type of balancing options set in the
                      structure*/
3 Timeout fixed bin(31), /* Timeout after which re-balancing might
```

```
3 BalanceOptions    fixed bin(31), /* interrupt application activity */
                   /* Balancing options set by the issuing
                   application*/
```

### Referências relacionadas

“MQCNO-Opções de conexão” na página 324

A estrutura MQCNO permite que o aplicativo especifique opções relacionadas à conexão com o gerenciador de filas.. A estrutura é um parâmetro de entrada / saída na chamada MQCONN.

### V 9.3.0 **StrucId (MQCHAR4) para MQBNO**

Este é o identificador da estrutura de opções de balanceamento. Ele é sempre um campo de entrada Seu valor inicial é BNO.

O valor deve ser:.

#### **BNO**

Identificador para a estrutura de opções de balanceamento.

Para a linguagem de programação C, a constante MQBNO\_STRUC\_ID\_ARRAY também é definida. Essa constante tem o mesmo valor que o BNO, mas é uma matriz de caracteres em vez de uma sequência.

Você deve fornecer um valor válido para **StrucId** ou MQRC\_BNO\_ERROR é retornado.

### V 9.3.0 **Versão (MQLONG) para MQBNO**

Este é o número da versão da estrutura de opções de balanceamento Ele é sempre um campo de entrada

O valor deve ser:.

#### **MQBNO\_VERSION\_1**

Número da versão da estrutura de opções de balanceamento.

Você deve fornecer um valor válido para **Versão** ou MQRC\_BNO\_ERROR é retornado.

### V 9.3.0 **ApplicationType (MQLONG) para MQBNO**

O tipo de conjunto de opções de balanceamento na estrutura

Os valores possíveis são:

#### **MQBNO\_BALTYPE\_SIMPLE**

Balanceamento simples; nenhuma regra específica é aplicada além daquelas descritas em [Rebalanceamento de aplicativo de influenciamento em clusters uniformes](#).

#### **MQBNO\_BALTYPE\_REQREP**

Balanceamento de solicitação/resposta; após cada chamada MQPUT, espera-se uma chamada MQGET correspondente para uma mensagem de resposta. O balanceamento é atrasado até que uma mensagem desse tipo seja recebida ou que a configuração EXPIRY da mensagem de solicitação seja excedida.

Isto é sempre um campo de entrada. O valor inicial desse campo é MQBNO\_BALTYPE\_SIMPLE.

Você deve fornecer um valor apenas para o campo **ApplicationType** ou MQRC\_BNO\_ERROR é retornado.

**Nota:** Um valor adicional para esse campo de MQBNO\_BALTYPE\_RA\_MANAGED é reservado para uso pelo Adaptador de Recursos IBM MQ para ambientes JEE. Embora seja um erro para um aplicativo fornecer esse valor diretamente ele pode, por exemplo, ser relatado ao consultar o status do aplicativo.

### V 9.3.0 **Tempo limite (MQLONG) para MQBNO**

O **Timeout** após o qual o rebalanceamento pode interromper a atividade do aplicativo

Os valores possíveis são:

#### **MQBNO\_TIMEOUT\_AS\_DEFAULT**

O valor de tempo limite padrão do conjunto.



### **MQBNO\_TIMEOUT\_IMMEDIATE**

Ocorre um tempo limite imediato

### **MQBNO\_TIMEOUT\_NEVER**

Não ocorre nenhum tempo limite

O valor inicial desse campo é MQBNO\_TIMEOUT\_AS\_DEFAULT.

Deve-se fornecer um valor apenas dos valores definidos ou um valor de 0-999999999 segundos para o campo **Timeout** ou MQRC\_BNO\_ERROR é retornado.

## **V9.3.0 BalanceOptions (MQLONG) para MQBNO**

As opções de balanceamento configuradas pelo aplicativo de emissão

Os valores possíveis são:

### **MQBNO\_OPTIONS\_NONE**

Nenhuma opção está definida

### **MQBNO\_OPTIONS\_IGNORE\_TRANS**

A configuração dessa opção permite o rebalanceamento de aplicativos mesmo durante uma transação.

O valor inicial deste campo é MQBNO\_OPTIONS\_NONE.

É possível fornecer qualquer combinação dos valores definidos usando a lógica ou o caractere para o campo **BalanceOptions**. Quaisquer valores que não forem válidos farão com que um MQRC\_BNO\_ERROR seja retornado.

## **MQBO-Opções de início de**

A estrutura MQBO permite que o aplicativo especifique opções relacionadas à criação de uma unidade de trabalho.. A estrutura é um parâmetro de entrada / saída na chamada MQBEGIN..

## **Disponibilidade**

A estrutura MQBO está disponível nas plataformas a seguir:

-  AIX
-  IBM i
-  Linux
-  Windows

A estrutura MQBO não está disponível para IBM MQ MQI clients.

## **Conjunto de caracteres e codificação**

Os dados no MQBO devem estar no conjunto de caracteres fornecido pelo atributo do gerenciador de filas do **CodedCharSetId** e pela codificação do gerenciador de fila local fornecido pelo MQENC\_NATIVE. No entanto, se o aplicativo estiver em execução como um cliente MQI MQ , a estrutura deverá estar no conjunto de caracteres e na codificação do cliente.

## **Campos**

**Nota:** Na tabela a seguir, os campos são agrupados por uso e não alfabeticamente. Os tópicos filhos seguem a mesma sequência.

Tabela 470. Campos em MQBO para MQBO

Nome e descrição do campo	Nome da constante	Valor inicial (se houver) da constante
StrucId (identificador de estrutura)	MQBO_STRUC_ID	'BO--'
Versão (número da versão da estrutura).	MQBO_VERSION_1	1
Opções (opções que controlam a ação de MQBEGIN)	MQBO_NONE	0

**Notas:**

1. O símbolo - representa um único caractere em branco.
2. Na linguagem de programação C, a variável macro MQBO\_DEFAULT contém os valores listados na tabela. Use-o da seguinte maneira para fornecer valores iniciais para os campos na estrutura:

```
MQBO MyBO = {MQBO_DEFAULT};
```

## Declarações de idiomas

### Declaração C para MQBO

```
typedef struct tagMQBO MQBO;
struct tagMQBO {
    MCHAR4  StrucId; /* Structure identifier */
    MQLONG  Version; /* Structure version number */
    MQLONG  Options; /* Options that control the action of MQBEGIN */
};
```

### Declaração COBOL para MQBO

```
** MQBO structure
10 MQBO.
** Structure identifier
15 MQBO-STRUCID PIC X(4).
** Structure version number
15 MQBO-VERSION PIC S9(9) BINARY.
** Options that control the action of MQBEGIN
15 MQBO-OPTIONS PIC S9(9) BINARY.
```

### Declaração PL/I para MQBO

```
dcl
  1 MQBO based,
  3 StrucId char(4), /* Structure identifier */
  3 Version fixed bin(31), /* Structure version number */
  3 Options fixed bin(31); /* Options that control the action of
                          MQBEGIN */
```

### Declaração Visual Basic para MQBO

```
Type MQBO
  StrucId As String*4 'Structure identifier'
  Version As Long 'Structure version number'
  Options As Long 'Options that control the action of MQBEGIN'
End Type
```

### **StrucId (MCHAR4) para MQBO**

Este é o identificador da estrutura de opções de início. Ele é sempre um campo de entrada Seu valor é MQBO\_STRUC\_ID.

O valor deve ser:.

### **MQBO\_STRUC\_ID**

Identificador para a estrutura de opções de início

Para a linguagem de programação C, a constante MQBO\_STRUC\_ID\_ARRAY também é definida. Isso tem o mesmo valor que MQBO\_STRUC\_ID, mas é uma matriz de caracteres em vez de uma cadeia.

### **Versão (MQLONG) para MQBO**

Este é o número da versão da estrutura de opções de início Ele é sempre um campo de entrada

O valor deve ser:.

### **MQBO\_VERSION\_1**

Número da versão para a estrutura de opções iniciais.

A constante a seguir especifica o número da versão atual:

### **MQBO\_CURRENT\_VERSION**

Versão atual da estrutura de opções iniciais.

### **Opções (MQLONG) para MQBO**

Esse campo é sempre um campo de entrada Seu valor inicial é MQBO\_NONE.

O valor deve ser:.

### **MQBO\_NONE**

Nenhuma opção especificada.

## **MQCBC-Contexto de retorno de chamada**

A estrutura MQCBC é usada para especificar informações de contexto que são transmitidas para uma função de retorno de chamada. A estrutura é um parâmetro de entrada / saída na chamada para uma rotina do consumidor da mensagem

## **Disponibilidade**

A estrutura MQCBC está disponível nas plataformas a seguir:

-  AIX
-  IBM i
-  Linux
-  Windows
-  z/OS

e para o IBM MQ MQI clients conectado a esses sistemas

## **Versão**

A versão atual do MQCBC é MQCBC\_VERSION\_2.

## **Conjunto de caracteres e codificação**

Os dados no MQCBC devem estar no conjunto de caracteres fornecido pelo atributo do gerenciador de filas do **CodedCharSetId** e pela codificação do gerenciador de filas locais fornecido pelo MQENC\_NATIVE No entanto, se o aplicativo estiver em execução como um cliente MQI MQ , a estrutura estará no conjunto de caracteres e na codificação do cliente.

## Campos

Não há valores iniciais para a estrutura MQCBC .. A estrutura é transmitida como um parâmetro para uma rotina de retorno de chamada O gerenciador de filas inicializa a estrutura; os aplicativos nunca inicializam.

### Notas:

- Na tabela a seguir, os campos são agrupados por uso e não alfabeticamente. Os tópicos filhos seguem a mesma sequência.
- Não há valores iniciais para a estrutura do MQCBC A estrutura é transmitida como um parâmetro para uma rotina de retorno de chamada O gerenciador de filas inicializa a estrutura; os aplicativos nunca inicializam.

Tabela 471. Campos em MQCBC	
Campo	Descrição
<u>StrucID</u>	Identificador de estruturação
<u>Versão</u>	Número de versão da estrutura
<u>CallType</u>	Por que a função foi chamada
<u>Hobj</u>	Manipulação de objetos
<u>CallbackArea</u>	Campo para a função de retorno de chamada a ser usada
<u>ConnectionArea</u>	Campo para a função de retorno de chamada a ser usada
<u>CompCode</u>	Código de conclusão
<u>Razão</u>	Código de razão
<u>Estado</u>	Indicação do estado do consumidor atual
<u>DataLength</u>	Comprimento da mensagem
<u>BufferLength</u>	Comprimento do buffer de mensagem em bytes
<u>Sinalizadores</u>	Sinalizadores gerais
<b>Nota:</b> O campo restante será ignorado se a Versão for menor que MQCBC_VERSION_2	
<u>ReconnectDelay</u>	Número de milissegundos antes da tentativa de reconexão

## Declarações de idiomas

Declaração C para MQCBC

```
typedef struct tagMQCBC MQCBC;
struct tagMQCBC {
    MQCHAR4   StrucId;           /* Structure identifier */
    MQLONG    Version;          /* Structure version number */
    MQLONG    CallType;         /* Why Function was called */
    MQHOBj    Hobj;             /* Object Handle */
    MQPTR     CallbackArea;     /* Callback data passed to the function */
    MQPTR     ConnectionArea;   /* MQCTL data area passed to the function */
    MQLONG    CompCode;         /* Completion Code */
    MQLONG    Reason;           /* Reason Code */
    MQLONG    State;            /* Consumer State */
    MQLONG    DataLength;       /* Message Data Length */
    MQLONG    BufferLength;      /* Buffer Length */
    MQLONG    Flags;            /* Flags containing information about
                                this consumer */

    /* Ver:1 */
    MQLONG    ReconnectDelay;   /* Number of milliseconds before */
    /* Ver:2 */ };              /* reconnect attempt */
```

## Declaração COBOL para MQCBC

```
** MQCBC structure
10 MQCBC.
** Structure Identifier
15 MQCBC-STRUCID PIC X(4).
** Structure Version
15 MQCBC-VERSION PIC S9(9) BINARY.
** Call Type
15 MQCBC-CALLTYPE PIC S9(9) BINARY.
** Object Handle
15 MQCBC-HOBJ PIC S9(9) BINARY.
** Callback User Area
15 MQCBC-CALLBACKAREA POINTER
** Connection Area
15 MQCBC-CONNECTIONAREA POINTER
** Completion Code
15 MQCBC-COMPCODE PIC S9(9) BINARY.
** Reason Code
15 MQCBC-REASON PIC S9(9) BINARY.
** Consumer State
15 MQCBC-STATE PIC S9(9) BINARY.
** Data Length
15 MQCBC-DATALENGTH PIC S9(9) BINARY.
** Buffer Length
15 MQCBC-BUFFERLENGTH PIC S9(9) BINARY.
** Flags
15 MQCBC-FLAGS PIC S9(9) BINARY.
** Ver:1 **
** Number of milliseconds before reconnect attempt
15 MQCBC-RECONNECTDELAY PIC S9(9) BINARY.
** Ver:2 **
```

## Declaração de PL/I para MQCBC

```
dcl
1 MQCBC based,
3 StrucId char(4), /* Structure identifier */
3 Version fixed bin(31), /* Structure version */
3 CallType fixed bin(31), /* Callback type */
3 Hobj fixed bin(31), /* Object Handle */
3 CallbackArea pointer, /* User area passed to the function */
3 ConnectionArea pointer, /* Connection User Area */
3 CompCode fixed bin(31); /* Completion Code */
3 Reason fixed bin(31); /* Reason Code */
3 State fixed bin(31); /* Consumer State */
3 DataLength fixed bin(31); /* Message Data Length */
3 BufferLength fixed bin(31); /* Message Buffer length */
3 Flags fixed bin(31); /* Consumer Flags */
/* Ver:1 */
3 ReconnectDelay fixed bin(31); /* Number of milliseconds before */
/* Ver:2 */ /* reconnect attempt */
```

## Declaração do High Level Assembler para MQCBC

```
MQCBC DSECT
MQCBC DS 0F Force fullword alignment
MQCBC_STRUCID DS CL4 Structure identifier
MQCBC_VERSION DS F Structure version number
MQCBC_CALLTYPE DS F Why Function was called
MQCBC_HOBJ DS F Object Handle
MQCBC_CALLBACKAREA DS A Callback data passed to the function
MQCBC_CONNECTIONAREA DS A MQCTL Data area passed to the function
MQCBC_COMPCODE DS F Completion Code
MQCBC_REASON DS F Reason Code
MQCBC_STATE DS F Consumer State
MQCBC_DATALENGTH DS F Message Data Length
MQCBC_BUFFERLENGTH DS F Buffer Length
MQCBC_FLAGS DS F Flags containing information about this consumer
MQCBC_RECONNECTDELAY DS F Number of milliseconds before reconnect
MQCBC_LENGTH EQU *-MQCBC
MQCBC_AREA DS CL(MQCBC_LENGTH)
```

### ***StrucId (MQCHAR4) para MQCBC***

Este é o identificador de estrutura da estrutura de contexto de retorno de chamada. Ele é sempre um campo de entrada. Seu valor é MQCBC\_STRUC\_ID.

O valor deve ser:.

#### **MQCBC\_STRUC\_ID**

Identificador para estrutura de contexto de retorno de chamada.

Para a linguagem de programação C, a constante MQCBC\_STRUC\_ID\_ARRAY também é definida. Ele tem o mesmo valor que MQCBC\_STRUC\_ID, mas é uma matriz de caracteres em vez de uma cadeia.

### ***Versão (MQLONG) para MQCBC***

Este é o número da versão da estrutura de contexto de retorno de chamada. Ele é sempre um campo de entrada.

O valor deve ser:.

#### **MQCBC\_VERSION\_1**

Estrutura de contexto de retorno de chamada da Versão 1.

A constante a seguir especifica o número da versão atual:

#### **MQCBC\_CURRENT\_VERSION**

A versão atual da estrutura de contexto de retorno de chamada.

A função de retorno de chamada é sempre passada a versão mais recente da estrutura...

### ***CallType (MQLONG) para MQCBC***

Campo contendo informações sobre por que essa função foi chamada; os valores a seguir são definidos.

Tipos de chamada de entrega de mensagem: esses tipos de chamada contêm informações sobre uma mensagem... Os parâmetros **DataLength** e **BufferLength** são válidos para esses tipos de chamada.

#### **MQCBCT\_MSG\_REMOVIDO**

A função de consumidor de mensagens foi chamada com uma mensagem que foi removida destrutivamente da manipulação de objetos.

Se o valor de *CompCode* for MQCC\_WARNING, o valor do campo *Reason* será MQRC\_TRUNCATED\_MSG\_ACCEPTED ou um dos códigos indicando um problema de conversão de dados.

#### **MQCBCT\_MSG\_NOT\_REMOVE**

A função de consumidor de mensagens foi chamada com uma mensagem que ainda não foi removida destrutivamente do identificador de objetos. A mensagem pode ser removida destrutivamente da manipulação de objetos usando o *MsgToken*.

A mensagem pode não ter sido removida porque:

- As opções MQGMO solicitaram uma operação de navegação, MQGMO\_BROWSE\_\*
- A mensagem é maior que o buffer disponível e as opções MQGMO não especificam MQGMO\_ACCEPT\_TRUNCATED\_MSG

Se o valor de *CompCode* for MQCC\_WARNING, o valor do campo *Reason* será MQRC\_TRUNCATED\_MSG\_FAILED ou um dos códigos indicando um problema de conversão de dados.

Tipos de chamada de controle de retorno de chamada: Esses tipos de chamada contêm informações sobre o controle do retorno de chamada e não contêm detalhes sobre uma mensagem.. Esses tipos de chamada são solicitados usando Opções na estrutura MQCBD.

Os parâmetros **DataLength** e **BufferLength** não são válidos para esses tipos de chamadas.

### **MQCBCT\_REGISTER\_CALL**

O propósito deste tipo de chamada é permitir que a função de retorno de chamada execute alguma configuração inicial..

A função de retorno de chamada é chamada imediatamente após o retorno de chamada ser registrado, ou seja, no retorno de uma chamada MQCB usando um valor para o campo *Operation* de MQOP\_REGISTER.

Esse tipo de chamada é usado para consumidores de mensagens e manipuladores de eventos.

Se solicitado, esta é a primeira chamada da função de retorno de chamada..

O valor do campo *Reason* é MQRC\_NONE.

### **MQCBCT\_START\_CALL**

O propósito desse tipo de chamada é permitir que a função de retorno de chamada execute alguma configuração quando ela for iniciada, por exemplo, restabelecendo os recursos que foram limpos quando foram interrompidos anteriormente

A função de retorno de chamada é chamada quando a conexão é iniciada usando MQOP\_START ou MQOP\_START\_WAIT..

Se uma função de retorno de chamada for registrada dentro de outra função de retorno de chamada, esse tipo de chamada será chamado quando o retorno de chamada retornar.

Esse tipo de chamada é usado apenas para consumidores da mensagem

O valor do campo *Reason* é MQRC\_NONE.

### **MQCBCT\_STOP\_CALL**

O propósito desse tipo de chamada é permitir que a função de retorno de chamada execute alguma limpeza quando ela for interrompida por um tempo, por exemplo, limpando recursos adicionais que foram obtidos durante o consumo de mensagens

A função de retorno de chamada é chamada quando uma chamada MQCTL é emitida usando um valor para o campo *Operation* de MQOP\_STOP.

Esse tipo de chamada é usado apenas para consumidores da mensagem

O valor do campo *Reason* é configurado para indicar o motivo da parada.

### **MQCBCT\_DEREGISTER\_CALL**

O propósito deste tipo de chamada é permitir que a função de retorno de chamada execute a limpeza final no final do processo de consumo... A função de retorno de chamada é chamada quando:

- A função de retorno de chamada tem o registro removido usando uma chamada MQCB com MQOP\_DEREGISTER.
- A fila é fechada, causando um cancelamento implícito. Nesta instância, a função de retorno de chamada é transmitida MQHO\_UNUSABLE\_HOBY como a manipulação de objetos.
- Chamada MQDISC concluída-causando um fechamento implícito e, portanto, um cancelamento de registro. Nesse caso, a conexão não será desconectada imediatamente e qualquer transação em andamento ainda não será confirmada

Se qualquer uma dessas ações for executada dentro da própria função de retorno de chamada, a ação será chamada quando o retorno de chamada for retornado.

Esse tipo de chamada é usado para consumidores de mensagens e manipuladores de eventos.

Se solicitado, esta é a última chamada da função de retorno de chamada.

O valor do campo *Reason* é configurado para indicar o motivo da parada.

### **MQCBCT\_EVENT\_CALL**

#### **Função do manipulador de eventos**

A função do manipulador de eventos foi chamada sem uma mensagem quando o gerenciador de filas ou a conexão para ou quiesce.

Essa chamada pode ser usada para executar a ação apropriada para todas as funções de retorno de chamada

### **Função do consumidor de mensagens**

A função do consumidor de mensagens foi chamada sem uma mensagem quando um erro (*CompCode* = MQCC\_FAILED) foi detectado que é específico para a manipulação de objetos; por exemplo, *Reason code* = MQRC\_GET\_INHIBITED..

O valor do campo *Reason* é configurado para indicar o motivo da chamada.

### **MQCBCT\_MC\_EVENT\_CALL**

A função do manipulador de eventos foi chamada para eventos multicast; o manipulador de eventos recebe eventos IBM MQ Multicast em vez de eventos 'normais' IBM MQ .

Para obter mais informações sobre MQCBCT\_MC\_EVENT\_CALL, consulte [Relatório de exceções multicast](#).

### **Hobj (MQHOBJ) para MQCBC**

Esta é a manipulação de objetos para as chamadas para o consumidor de mensagem

Para um manipulador de eventos, esse valor é MQHO\_NONE

O aplicativo pode usar esse identificador e o token de mensagem no bloco Obter Opções de Mensagem para obter a mensagem se uma mensagem não tiver sido removida da fila

Isto é sempre um campo de entrada. O valor inicial deste campo é MQHO\_UNUSABLE\_HOBJ

### **CallbackArea (MQPTR) para MQCBC**

Este campo está disponível para a função de retorno de chamada usar.

O gerenciador de filas não toma decisões com base no conteúdo desse campo e é transmitido inalterado a partir do campo [CallbackArea](#) na estrutura MQCBD, que é um parâmetro na chamada MQCB usado para definir a função de retorno de chamada.

Mudanças no *CallbackArea* são preservadas nas chamadas da função de retorno de chamada para um *Hobj*. Este campo não é compartilhado com funções de retorno de chamada para outros identificadores

Este é um campo de entrada / saída para a função callback. O valor inicial desse campo é um ponteiro nulo ou bytes nulos

### **ConnectionArea (MQPTR) para MQCBC**

Este campo está disponível para a função de retorno de chamada usar.

O gerenciador de filas não toma decisões com base no conteúdo desse campo e é transmitido inalterado a partir do campo [ConnectionArea](#) na estrutura MQCTLO, que é um parâmetro na chamada MQCTL usado para controlar a função de retorno de chamada.

Quaisquer mudanças feitas nesse campo pelas funções de retorno de chamada são preservadas nas chamadas da função de retorno de chamadas. Essa área pode ser usada para transmitir informações que devem ser compartilhadas pelas funções de retorno de chamada. Ao contrário do *CallbackArea*, essa área é comum em todos os retornos de chamada para uma manipulação de conexões

Este é um campo de entrada e de saída O valor inicial desse campo é um ponteiro nulo ou bytes nulos

### **CompCode (MQLONG) para MQCBC**

Este é o código de conclusão. Indica se houve problemas ao consumir a mensagem.

O valor é um dos seguintes:



**MQCC\_OK**

Conclusão bem-sucedida

**MQCC\_WARNING**

Aviso (conclusão parcial)

**MQCC\_FAILED**

Chamada com falha

Este é um campo de entrada. O valor inicial desse campo é MQCC\_OK..

**Razão (MQLONG) para MQCBC**Este é o código de razão que qualifica o *CompCode*

Este é um campo de entrada. O valor inicial desse campo é MQRC\_NONE.

**Estado (MQLONG) para MQCBC**

Uma indicação quanto ao estado do consumidor atual Este campo é de maior valor para um aplicativo quando um código de razão diferente de zero é transmitido para a função do consumidor...

É possível usar esse campo para simplificar a programação de aplicativos, pois não é necessário codificar o comportamento para cada código de razão

Este é um campo de entrada. O valor inicial desse campo é MQCS\_NONE

<i>Tabela 472.</i>		
<b>Estado</b>	<b>Ação do gerenciador de filas</b>	<b>Valor da constante</b>
<i>MQCS_NONE</i> Este código de razão representa uma chamada normal sem informações de razão adicionais	Nenhum; esta é a operação normal	0
<i>MQCS_SUSPENDED_TEMPORARY</i> Esses códigos de razão representam condições temporárias..	A rotina de retorno de chamada é chamada para relatar a condição e, em seguida, suspensa Após um período de tempo, o sistema pode tentar a operação novamente, o que pode levar à mesma condição sendo levantada novamente.	1
<i>MQCS_SUSPENDED_USER_ACTION</i> Esses códigos de razão representam condições nas quais o retorno de chamada precisa tomar ação para resolver a condição.	O consumidor é suspenso, e a rotina de retorno de chamada é chamada para relatar a condição A rotina de retorno de chamada deve resolver a condição se possível e RESUME ou fechar a conexão.	2
<i>MQCS_SUSPENDED</i> Esses códigos de razão representam falhas que evitam retornos de chamada de mensagens adicionais	O gerenciador de filas suspende automaticamente a função de retorno de chamada Se a função de retorno de chamada for retomada, provavelmente receberá o mesmo código de razão novamente.	3
<i>MQCS_STOPPED</i> Esses códigos de razão representam o término do consumo de mensagens	Entregue para o manipulador de exceções e para retornos de chamadas que especificaram MQCBDO_STOP_CALL Nenhuma mensagem adicional pode ser consumida	4

### **DataLength (MQLONG) para MQCBC**

Este é o comprimento em bytes dos dados do aplicativo na mensagem. Se o valor for zero, significa que a mensagem não contém dados do aplicativo.

O campo DataLength contém o comprimento da mensagem, mas não necessariamente o comprimento dos dados da mensagem transmitidos ao consumidor. Pode ser que a mensagem tenha sido truncada. Use o campo ReturnedLength no MQGMO para determinar quantos dados foram realmente passados para o consumidor.

Se o código de razão indicar que a mensagem foi truncada, é possível usar o campo DataLength para determinar o tamanho da mensagem real. Isso permite determinar o tamanho do buffer necessário para acomodar os dados da mensagem e, em seguida, emitir uma chamada MQCB para atualizar o MaxMsgLength com um valor apropriado.

Se a opção MQGMO\_CONVERT for especificada, a mensagem convertida poderá ser maior que o valor retornado para DataLength. Nesses casos, o aplicativo provavelmente precisa emitir uma chamada MQCB para atualizar o MaxMsgLength para ser maior que o valor retornado pelo gerenciador de filas para DataLength.

Para evitar problemas de truncamento de mensagens, especifique MaxMsg como MQCBD\_FULL\_MSG\_LENGTH. Isso faz com que o gerenciador de filas aloque um buffer para o comprimento da mensagem integral após a conversão dos dados. No entanto, esteja ciente de que, mesmo se essa opção for especificada, ainda é possível que armazenamento suficiente não esteja disponível para processar corretamente a solicitação. Os aplicativos devem sempre verificar o código de razão retornado. Por exemplo, se não for possível alocar armazenamento suficiente para converter a mensagem, as mensagens serão retornadas para o aplicativo não convertidas.

Este é um campo de entrada para a função do consumidor de mensagens; não é relevante para uma função do manipulador de eventos.

### **BufferLength (MQLONG) para MQCBC**

Este campo é o comprimento em bytes do buffer de mensagem que foi transmitido para esta função.

O buffer pode ser maior que o valor de Comprimento MaxMsg definido para o consumidor e o valor ReturnedLength no MQGMO.

O comprimento da mensagem real é fornecido no campo DataLength ..

O aplicativo pode usar o buffer inteiro para seus próprios propósitos durante a função de retorno de chamada.

Este é um campo de entrada para a função do consumidor de mensagens; não é relevante para uma função do manipulador de exceções.

### **Sinalizadores (MQLONG) para MQCBC**

Sinalizadores que contêm informações sobre esse consumidor

A seguinte opção é definida:

#### **MQCBCF\_READA\_BUFFER\_EMPTY**

Esse sinalizador poderá ser retornado se uma chamada MQCLOSE anterior usando a opção MQCO QUIESCE falhar com um código de razão de MQRC\_READ\_AHEAD\_MSGS

Este código indicou que a última mensagem de leitura antecipada está sendo retornada e que o buffer agora está vazio. Se o aplicativo emitir outra chamada MQCLOSE usando a opção MQCO QUIESCE, ele será bem-sucedido.

Observe que não é garantido que um aplicativo receberá uma mensagem com esse sinalizador configurado, pois ainda pode haver mensagens no buffer de leitura antecipada que não correspondem aos critérios de seleção atuais. Nesta instância, a função do consumidor é chamado com o código de razão MQRC\_HOBJ QUIESCED.

Se o buffer de leitura antecipada estiver completamente vazio, o consumidor será chamado com a sinalização MQCBCF\_READA\_BUFFER\_EMPTY e o código de razão MQRC\_HOBJ QUIESCED\_NO\_MSGS.

Este é um campo de entrada para a função do consumidor de mensagens; não é relevante para uma função do manipulador de eventos.

### **ReconnectDelay (MQLONG) para MQCBC**

ReconnectDelay indica quanto tempo o gerenciador de filas esperará antes de tentar se reconectar. O campo pode ser modificado por um manipulador de eventos para alterar o atraso ou parar a reconexão.

Use o campo ReconnectDelay somente se o valor do campo Motivo no Contexto de Retorno de Chamada for MQRC\_RECONNECTING.

Na entrada para o manipulador de eventos, o valor de ReconnectDelay é o número de milissegundos que o gerenciador de filas aguardará antes de fazer uma tentativa de reconexão.. Tabela 473 na página 295 lista os valores que podem ser configurados para modificar o comportamento do gerenciador de filas no retorno do manipulador de eventos

<i>Tabela 473. Valores de ReconnectDelay</i>		
<b>Nome</b>	<b>Value</b>	<b>Descrição</b>
MQRD_NO_RECONNECT	-1	Não faça mais tentativas de reconexão. Um erro é retornado para o aplicativo
MQRD_NO_DELAY	0	Tente reconectar imediatamente.
<i>Milliseconds</i>	>0	Aguarde por esses milissegundos antes de tentar novamente a conexão.

### **MQCBD-Descriptor de retorno de chamada**

A estrutura MQCBD é usada para especificar uma função de retorno de chamadas e as opções que controlam seu uso pelo gerenciador de fila.. A estrutura é um parâmetro de entrada na chamada MQCB

### **Disponibilidade**

A estrutura MQCBD está disponível nas plataformas a seguir:

-  AIX
-  IBM i
-  Linux
-  Windows
-  z/OS

e para o IBM MQ MQI clients conectado a esses sistemas

### **Versão**

A versão atual do MQCBD é MQCBD\_VERSION\_1

### **Conjunto de caracteres e codificação**

Os dados em MQCBD devem estar no conjunto de caracteres fornecido pelo atributo do gerenciador de filas do **CodedCharSetId** e pela codificação do gerenciador de fila local fornecido por MQENC\_NATIVE. No entanto, se o aplicativo estiver em execução como um cliente MQI MQ , a estrutura deverá estar no conjunto de caracteres e na codificação do cliente.

## Campos

**Nota:** Na tabela a seguir, os campos são agrupados por uso e não alfabeticamente. Os tópicos filhos seguem a mesma sequência.

<i>Tabela 474. Campos em MQCBD</i>		
Nome e descrição do campo	Nome da constante	Valor inicial (se houver) da constante
<u>StrucID</u> (identificador de estruturas).	MQCBD_STRUC_ID	'CBD~'
<u>Versão</u> (número da versão da estrutura).	MQCBD_VERSION_1	1
<u>CallbackType</u> (tipo de função de callback)	MQCBT_MESSAGE_CONSUMER	1
<u>Opções</u> (opções controlando o consumo de mensagens).	MQCBDO_NONE	0
<u>CallbackArea</u> (campo para a função de retorno de chamada usar)	Nenhum	Ponteiro nulo ou espaços em branco nulos
<u>CallbackFunction</u> (se a função é chamada como uma chamada API)	Nenhum	Ponteiro nulo ou espaços em branco nulos
<u>CallbackName</u> (se a função é chamada como um programa dinamicamente vinculado)	Nenhum	Cadeia nula ou espaços em branco
<u>MaxMsgComprimento</u> (comprimento da mensagem mais longa que pode ser lida)	MQCBD_FULL_MSG_LENGTH	-1

**Notas:**

1. O símbolo ~ representa um único caractere em branco.
2. O valor Sequência nula ou espaços em branco indica a sequência nula na linguagem de programação C e caracteres em branco em outras linguagens de programação.
3. Na linguagem de programação C, a variável macroMQCBD\_DEFAULT contém os valores que são listados na tabela. Use-o da seguinte maneira para fornecer valores iniciais para os campos na estrutura:

```
MQCBD MyCBD = {MQCBD_DEFAULT};
```

## Declarações de idiomas

Declaração C para MQCBD

```
typedef struct tagMQCBD MQCBD;
struct tagMQCBD {
    MQCHAR4    StructId;           /* Structure identifier */
    MQLONG     Version;           /* Structure version number */
    MQLONG     CallbackType;      /* Callback function type */
    MQLONG     Options;           /* Options controlling message
                                consumption */
    MQPTR      CallbackArea;      /* User data passed to the function */
    MQPTR      CallbackFunction;  /* Callback function pointer */
    MQCHAR128  CallbackName;     /* Callback name */
    MQLONG     MaxMsgLength;     /* Maximum message length */
};
```

## Declaração COBOL para MQCBD

```
** MQCBCD structure
10 MQCBCD.
** Structure Identifier
15 MQCBCD-STRUCID PIC X(4).
** Structure Version
15 MQCBCD-VERSION PIC S9(9) BINARY.
** Callback Type
15 MQCBCD-CALLBACKTYPE PIC S9(9) BINARY.
** Options
15 MQCBCD-OPTIONS PIC S9(9) BINARY.
** Callback User Area
15 MQCBCD-CALLBACKAREA POINTER
** Callback Function Pointer
15 MQCBCD-CALLBACKFUNCTION FUNCTION-POINTER
** Callback Program Name
15 MQCBCD-CALLBACKNAME PIC X(128)
** Maximum Message Length
15 MQCBCD-MAXMSGLENGTH PIC S9(9) BINARY.
```

## Declaração PL/I para MQCBD

```
dcl
1 MQCBCD based,
3 StructId char(4), /* Structure identifier*/
3 Version fixed bin(31), /* Structure version*/
3 CallbackType fixed bin(31), /* Callback function type */
3 Options fixed bin(31), /* Options */
3 CallbackArea pointer, /* User area passed to the function */
3 CallbackFunction pointer, /* Callback Function Pointer */
3 CallbackName char(128), /* Callback Program Name */
3 MaxMsgLength fixed bin(31); /* Maximum Message Length */
```

### **StrucId (MQCHAR4) para MQCBD**

Esse é o identificador da estrutura do descritor de retorno de chamada. Ele é sempre um campo de entrada. Seu valor é MQCBCD\_STRUC\_ID.

O valor deve ser:.

#### **MQCBCD\_STRUC\_ID**

Identificador para a estrutura do descritor de retorno de chamada

Para a linguagem de programação C, a constante MQCBCD\_STRUC\_ID\_ARRAY também é definida. Ele tem o mesmo valor que MQCBCD\_STRUC\_ID, mas é uma matriz de caracteres em vez de uma sequência.

### **Versão (MQLONG) para MQCBD**

Este é o número da versão da estrutura do descritor de retorno de chamada. Ele é sempre um campo de entrada.

O valor deve ser:.

#### **MQCBCD\_VERSION\_1**

Estrutura do descritor de retorno de chamada da versão 1

A constante a seguir especifica o número da versão atual:

#### **MQCBCD\_CURRENT\_VERSION**

Versão atual da estrutura do descritor de callback.

### **CallbackType (MQLONG) para MQCBD**

Estrutura do descritor de retorno de chamada-campo CallbackType

Este é o tipo da função de retorno de chamada.. O valor deve ser um de:

#### **MQCBT\_MESSAGE\_CONSUMER**

Define esse retorno de chamada como uma função do consumidor de mensagens..

Uma função de retorno de chamada do consumidor de mensagens é chamada quando uma mensagem, atendendo aos critérios de seleção especificados, está disponível em um identificador de objeto e a conexão é iniciada.

### **MQCBT\_EVENT\_HANDLER**

Define esse retorno de chamada como a rotina de evento assíncrono; ele não é orientado a consumir mensagens para um identificador.

O *Hobj* não é necessário na chamada MQCB que define o manipulador de eventos e será ignorado se especificado

O manipulador de eventos é chamado para condições que afetam todo o ambiente do consumidor de mensagens. A função do consumidor é chamada sem uma mensagem quando ocorre um evento, por exemplo, um gerenciador de filas ou parada de conexão ou quiesce. Ele não é chamado para condições específicas para um único consumidor de mensagens, por exemplo, MQRC\_GET\_INHIBITED.

Os eventos são entregues para o aplicativo, independentemente de a conexão ser iniciada ou interrompida, exceto nos ambientes a seguir:

- CICS no ambiente z/OS
- aplicativos não encadeados

Se o responsável pela chamada não passar um desses valores, a chamada falhará com um código *Reason* de MQRC\_CALLBACK\_TYPE\_ERROR

Isto é sempre um campo de entrada. O valor inicial desse campo é MQCBT\_MESSAGE\_CONSUMER.

### **Opções (MQLONG) para MQCBD**

Estrutura do Descritor de Retorno de Chamada-Campo Opções

É possível especificar uma ou mais dessas opções. Para especificar mais de uma opção, inclua os valores juntos (não inclua a mesma constante mais de uma vez) ou combine os valores usando a operação OR bit a bit (se a linguagem de programação suportar operações de bit).

#### **MQCBDO\_FAIL\_IF QUIESCING**

A chamada MQCB falhará se o gerenciador de filas estiver no estado de quiesce

No z/OS, essa opção também força a chamada MQCB a falhar se a conexão (para um aplicativo CICS ou IMS) estiver no estado de quiesce

Especifique MQGMO\_FAIL\_IF QUIESCING, nas opções MQGMO transmitidas na chamada MQCB, para causar notificação aos consumidores de mensagens quando eles estiverem quiesce.

**Opções de controle:** As opções a seguir controlam se a função de retorno de chamada é chamada, sem uma mensagem, quando o estado do consumidor é alterado:

#### **MQCBDO\_REGISTER\_CALL**

A função callback é chamada com o tipo de chamada MQCBCT\_REGISTER\_CALL.

#### **MQCBDO\_START\_CALL**

A função de retorno de chamada é chamada com o tipo de chamada MQCBCT\_START\_CALL

#### **MQCBDO\_STOP\_CALL**

A função de retorno de chamada é chamada com o tipo de chamada MQCBCT\_STOP\_CALL.

#### **MQCBDO\_DEREGISTER\_CALL**

A função callback é chamada com o tipo de chamada MQCBCT\_DEREGISTER\_CALL.

#### **MQCBDO\_EVENT\_CALL**

A função callback é chamada com o tipo de chamada MQCBCT\_EVENT\_CALL.

#### **MQCBDO\_MC\_EVENT\_CALL**

A função de retorno de chamada é chamada com o tipo de chamada MQCBCT\_MC\_EVENT\_CALL.

Consulte [CallType](#) para obter detalhes adicionais sobre esses tipos de chamada.

**Opção padrão:** se você não precisar de nenhuma das opções descritas, use a seguinte opção:

## **MQCBDO\_NONE**

Use esse valor para indicar que nenhuma outra opção foi especificada. Todas as opções assumem seus valores-padrão.

MQCBDO\_NONE é definido para auxiliar a documentação do programa; não se pretende que essa opção seja usada com nenhuma outra, mas como seu valor é zero, tal uso não pode ser detectado.

Este é um campo de entrada. O valor inicial do campo *Options* é MQCBDO\_NONE.

## **CallbackArea (MQPTR) para MQCBD**

Estrutura do descritor de retorno de chamada-campo CallbackArea

Este é um campo que está disponível para a função de retorno de chamada usar.

O gerenciador de filas não toma decisões com base no conteúdo desse campo e é transmitido inalterado a partir do campo CallbackArea na estrutura MQCBC, que é um parâmetro na declaração da função de retorno de chamada.

O valor é usado apenas em um *Operation* que possui um valor MQOP\_REGISTER, sem retorno de chamada definido atualmente, ele não substitui uma definição anterior.

Este é um campo de entrada e saída para a função de retorno de chamada. O valor inicial desse campo é um ponteiro nulo ou bytes nulos

## **CallbackFunction (MQPTR) para MQCBD**

Estrutura do descritor de retorno de chamada-campo CallbackFunction


A função de retorno de chamadas é chamada como uma chamada de função.

Use este campo para especificar um ponteiro para a função de retorno de chamada..

Você deve especificar *CallbackFunction* ou *CallbackName*. Se você especificar ambos, o código de razão MQRC\_CALLBACK\_ROUTINE\_ERROR será retornado..

Se nem *CallbackName* nem *CallbackFunction* for configurado, a chamada falhará com o código de razão MQRC\_CALLBACK\_ROUTINE\_ERROR

Essa opção não é suportada no seguinte ambiente: linguagens de programação e compiladores que não suportam referências de ponteiro de função. Nessas situações, a chamada falha com o código de razão MQRC\_CALLBACK\_ROUTINE\_ERROR.

 No z/OS, a função deve esperar ser chamada com convenções de ligação do S.O. Por exemplo, na linguagem de programação C, especifique:

```
#pragma linkage(MQCB_FUNCTION,OS)
```

Este é um campo de entrada. O valor inicial desse campo é um ponteiro nulo ou bytes nulos

**Nota:** Ao usar CICS com IBM WebSphere MQ 7.0.1, o consumo assíncrono será suportado se:

- Apar PK66866 é aplicado ao CICS TS 3.2
- Apar PK89844 é aplicado ao CICS TS 4.1

## **CallbackName (MQCHAR128) para MQCBD**

Estrutura do descritor de retorno de chamada-campo CallbackName

A função callback é chamada como um programa dinamicamente vinculado.

Você deve especificar *CallbackFunction* ou *CallbackName*. Se você especificar ambos, o código de razão MQRC\_CALLBACK\_ROUTINE\_ERROR será retornado..

Se nem *CallbackName* nem *CallbackFunction* não for configurado, a chamada falhará com o código de razão MQRC\_CALLBACK\_ROUTINE\_ERROR

O módulo é carregado quando a primeira rotina de retorno de chamada a ser usada é registrada e descarregado quando a última rotina de retorno de chamada para usá-la remove o registro.

Exceto onde indicado no texto a seguir, o nome é justificado à esquerda dentro do campo, sem espaços em branco incorporados; o próprio nome é preenchido com espaços em branco para o comprimento do campo. Nas descrições a seguir, os colchetes ([]) denotam informações opcionais:

### **IBM i**

O nome do retorno de chamadas pode ser um dos seguintes formatos:

- Programa de Biblioteca "/"
- Biblioteca "/" ServiceProgram ("FunctionName")

Por exemplo, MyLibrary/MyProgram(MyFunction).

O nome da biblioteca pode ser \*LIBL Ambos os nomes de biblioteca e de programa são limitados a um máximo de 10 caracteres.

### **AIX and Linux**

O nome do retorno de chamada é o nome de um módulo ou biblioteca carregável dinamicamente, sufixado com o nome de uma função residente nessa biblioteca. O nome da função deve estar entre parênteses. Opcionalmente, o nome da biblioteca pode ser prefixado com um caminho do diretório:

```
[path]library(function)
```

Se o caminho não for especificado, o caminho da procura do sistema será usado

O nome é limitado a no máximo 128 caracteres.

### **Windows**

O nome do retorno de chamada é o nome de uma biblioteca de link dinâmico, sufixado com o nome de uma função residente nessa biblioteca. O nome da função deve ser colocado entre parênteses. O nome da biblioteca pode, opcionalmente, ser prefixado com um caminho de diretório e uma unidade:

```
[d:][path]library(function)
```

Se a unidade e o caminho não forem especificados, o caminho da procura do sistema será usado

O nome é limitado a no máximo 128 caracteres.

### **z/OS**

O nome do retorno de chamada é o nome de um módulo de carregamento que é válido para a especificação no parâmetro EP da macro LINK ou LOAD

O nome é limitado a um máximo de 8 caracteres.

### **z/OS CICS**

O nome do retorno de chamada é o nome de um módulo de carregamento válido para especificação no parâmetro PROGRAM da macro do comando EXEC CICS LINK.

O nome é limitado a um máximo de 8 caracteres.

O programa pode ser definido como remoto utilizando a opção REMOTESYTEM da definição PROGRAM instalada ou pelo programa de roteamento dinâmico.

A região CICS remota deve ser conectada ao IBM MQ se o programa for usar chamadas API do IBM MQ . Observe, no entanto, que o campo Hobj na estrutura MQCBC não é válido em um sistema remoto.

Se ocorrer uma falha ao tentar carregar *CallbackName*, um dos códigos de erro a seguir será retornado para o aplicativo:

- MQRC\_MODULE\_NOT\_FOUND
- MQRC\_MODULE\_INVALID



- MQRC\_MODULE\_ENTRY\_NOT\_FOUND

Uma mensagem também é gravada no log de erros que contém o nome do módulo para o qual o carregamento foi tentado e o código de razão com falha do sistema operacional

Este é um campo de entrada. O valor inicial deste campo é uma cadeia nula ou espaços em branco.

### **MaxMsgComprimento (MQLONG) para MQCBD**

Este é o comprimento em bytes da mensagem mais longa que pode ser lida a partir do identificador e fornecida à rotina de retorno de chamada. Estrutura do descritor de retorno de chamada-campo Comprimento MaxMsg

Se uma mensagem tiver um comprimento maior, a rotina de retorno de chamada receberá *MaxMsgLength* bytes da mensagem e código de razão:

- MQRC\_TRUNCATED\_MSG\_FAILED ou
- MQRC\_TRUNCATED\_MSG\_ACCEPTED se tiver especificado MQGMO\_ACCEPT\_TRUNCATED\_MSG.

O comprimento da mensagem real é fornecido no campo DataLength da estrutura MQCBC

O seguinte valor especial é definido:

#### **MQCBD\_FULL\_MSG\_LENGTH**

O comprimento do buffer é ajustado pelo sistema para retornar as mensagens sem truncamento.

Se memória insuficiente estiver disponível para alocar um buffer para receber a mensagem, o sistema chama a função de retorno de chamada com um código de razão MQRC\_STORAGE\_NOT\_AVAILABLE.

Se, por exemplo, você solicitar a conversão de dados e houver memória insuficiente disponível para converter os dados da mensagem, a mensagem não convertida será transmitida para a função de retorno de chamada.

Este é um campo de entrada. O valor inicial do campo *MaxMsgLength* é MQCBD\_FULL\_MSG\_LENGTH.

### **MQCHARV-Sequência de Comprimento Variável**

Use a estrutura MQCHARV para descrever uma sequência de comprimento variável.

#### **Disponibilidade**

A estrutura MQCHARV está disponível nas plataformas a seguir:

-  AIX
-  IBM i
-  Linux
-  Windows

e para o IBM MQ MQI clients conectado a esses sistemas

#### **Conjunto de caracteres e codificação**

Os dados no MQCHARV devem estar na codificação do gerenciador de fila local fornecido por MQENC\_NATIVE e o conjunto de caracteres do campo VSCCSID dentro da estrutura. Se o aplicativo estiver em execução como um cliente MQ, a estrutura deverá estar na codificação do cliente.. Alguns conjuntos de caracteres possuem uma representação que depende da codificação.. Se VSCCSID for um desses conjuntos de caractere, a codificação usada será a mesma codificação dos outros campos no MQCHARV. O conjunto de caracteres identificado por VSCCSID pode ser um conjunto de caracteres de duplo byte (DBCS).

## Uso

A estrutura MQCHARV endereça dados que podem ser descontíguos com a estrutura que os contém. Para tratar esses dados, campos declarados com o tipo de dados do ponteiro podem ser utilizados. Esteja ciente de que COBOL não suporta o tipo de dados do ponteiro em todos os ambientes. Por isso, os dados também podem ser endereçados usando campos que contêm o deslocamento dos dados do início da estrutura que contém o MQCHARV.

## Campos

**Nota:** Na tabela a seguir, os campos são agrupados por uso e não alfabeticamente. Os tópicos filhos seguem a mesma sequência.

Tabela 475. Campos em MQCHARV		
Nome e descrição do campo	Nome da constante	Valor inicial (se houver) da constante
VSPtr (ponteiro para a sequência de comprimento variável)	Nenhum	Ponteiro nulo ou bytes nulos.
VSOOffset (deslocamento em bytes da sequência de comprimento variável do início da estrutura que contém essa estrutura MQCHARV)	Nenhum	0
VSBufSize (tamanho em bytes do buffer endereçado pelo campo VSPtr ou VSOOffset)	MQVS_USE_VSLENGTH	0
VSLength (comprimento em bytes da sequência de comprimento variável endereçada pelo campo VSPtr ou VSOOffset)	Nenhum	0
VSCCSID (identificador do conjunto de caracteres da sequência de comprimento variável endereçada pelo campo VSP ou VSOOffset)	MQCCSI_APPL	-3

**Nota:** Na linguagem de programação C, a variável de macro MQCHARV\_DEFAULT contém os valores listados na tabela. Ele pode ser usado da seguinte maneira para fornecer valores iniciais para os campos na estrutura:

```
MQCHARV MyVarStr = {MQCHARV_DEFAULT};
```

## Declarações de idiomas

Declaração C para MQCHARV

```
typedef struct tagMQCHARV MQCHARV;
struct tagMQCHARV {
    MQPTR    VSPtr;           /* Address of variable length string */
    MQLONG   VSOOffset;      /* Offset of variable length string */
    MQLONG   VSBufSize;     /* Size of buffer */
    MQLONG   VSLength;      /* Length of variable length string */
    MQLONG   VSCCSID;       /* CCSID of variable length string */
};
```

Declaração COBOL para MQCHARV

```
** MQCHARV structure
10  MQCHARV.
** Address of variable length string
15  MQCHARV-VSPTR          POINTER.
** Offset of variable length string
```

```

15 MQCHARV-VSOFFSET PIC S9(9) BINARY.
** Size of buffer
15 MQCHARV-VSBUFSIZE PIC S9(9) BINARY.
** Length of variable length string
15 MQCHARV-VSLENGTH PIC S9(9) BINARY.
** CCSID of variable length string
15 MQCHARV-VSCCSID PIC S9(9) BINARY.

```

**Nota:** Se você deseja portar um aplicativo COBOL entre ambientes, deverá descobrir se o tipo de dados do ponteiro está disponível em todos os ambientes desejados. Caso contrário, o aplicativo deverá endereçar os dados usando os campos de deslocamento em vez dos campos de ponteiro. Em ambientes nos quais os ponteiros não são suportados, é possível declarar os campos do ponteiro como sequências de bytes do comprimento apropriado, com o valor inicial sendo a sequência de bytes all-null. Não altere esse valor inicial se você estiver usando os campos de deslocamento. Uma maneira de fazer isso sem alterar os copybooks fornecidos é usar o seguinte:

```
COPY CMQCHRVV REPLACING POINTER BY ==BINARY PIC S9(9)==.
```

em que CMQCHRVV pode ser trocado para o copybook a ser usado

Declaração PL/I para MQCHARV

```

dcl
1 MQCHARV based,
3 VSPtr pointer, /* Address of variable length string */
3 VSOffset fixed bin(31), /* Offset of variable length string */
3 VSBufSize fixed bin(31), /* Size of buffer */
3 VSLength fixed bin(31), /* Length of variable length string */
3 VSCCSID fixed bin(31); /* CCSID of variable length string */

```

Declaração do High Level Assembler para MQCHARV

```

MQCHARV          DSECT
MQCHARV_VSPTR    DS F    Address of variable length string
MQCHARV_VSOFFSET DS F    Offset of variable length string
MQCHARV_VSBUFSIZE DS F    Size of buffer
MQCHARV_VSLENGTH DS F    Length of variable length string
MQCHARV_VSCCSID  DS F    CCSID of variable length string
*
MQCHARV_LENGTH   EQU *-MQCHARV
MQCHARV_AREA     DS    CL(MQCHARV_LENGTH)

```

### ***VSPtr (MQPTR) para MQCHARV***

Este é um ponteiro para a cadeia de comprimento variável..

É possível usar o campo VSPtr ou VSOffset para especificar a sequência de comprimento variável, mas não ambos.

O valor inicial desse campo é um ponteiro nulo ou bytes nulos

### ***VSOffset (MQLONG) para MQCHARV***

O deslocamento pode ser positivo ou negativo. É possível usar o campo VSPtr ou VSOffset para especificar a sequência de comprimento variável, mas não ambos. O deslocamento em bytes da sequência de comprimento variável do início do MQCHARV ou da estrutura que o contém.

Quando a estrutura MQCHARV é integrada em outra estrutura, esse valor é o deslocamento em bytes da sequência de comprimento variável do início da estrutura que contém essa estrutura MQCHARV. Quando a estrutura MQCHARV não está integrada dentro de outra estrutura, por exemplo, se ela for especificada como um parâmetro em uma chamada de função, o deslocamento será relativo ao início da estrutura MQCHARV

O valor inicial desse campo é 0.

### ***VBufSize (MQLONG) para MQCHARV***

Esse é o tamanho em bytes do buffer endereçado pelo campo VSPtr ou VSOffset.

Quando a estrutura MQCHARV é usada como um campo de saída em uma chamada de função, esse campo deve ser inicializado com o comprimento do buffer fornecido.. Se o valor de VSLength for maior que VBufSize , apenas VBufSize bytes de dados serão retornados ao responsável pela chamada no buffer.

Esse valor deve ser um valor maior ou igual a zero ou o seguinte valor especial que é reconhecido:

#### **MQVS\_USE\_VSLENGTH**

Quando especificado, o comprimento do buffer é obtido do campo VSLength na estrutura MQCHARV. Não use esse valor ao usar a estrutura como um campo de saída e um buffer será fornecido

Esse é o valor inicial desse campo.

### ***VSLength (MQLONG) para MQCHARV***

O comprimento em bytes da sequência de comprimento variável endereçada pelo campo VSPtr ou VSOffset.

O valor inicial desse campo é 0. O valor deve ser maior ou igual a zero ou o valor especial a seguir que é reconhecido:

#### **MQVS\_NULL\_TERMINATED**

Se MQVS\_NULL\_TERMINATED não for especificado, os bytes VSLength serão incluídos como parte da sequência. Se caracteres nulos estiverem presentes, eles não delimitarão a sequência.

Se MQVS\_NULL\_TERMINATED for especificado, a sequência será delimitada pelo primeiro nulo encontrado na sequência. O próprio nulo não é incluído como parte dessa cadeia.

**Nota:** O caractere nulo usado para finalizar uma sequência se MQVS\_NULL\_TERMINATE for especificado é um nulo do conjunto de códigos especificado por VSCCSID.


Por exemplo, em UTF-16 (CCSIDs 1200, 13488 e 17584), esta é a codificação Unicode de dois bytes em que um nulo é representado por um número de 16 bits de todos os zeros. Em UTF-16 , é comum localizar bytes únicos configurados para todos os zero que fazem parte de caracteres (caracteres ASCII de 7 bits, por exemplo), mas as sequências serão terminadas nulas apenas quando dois bytes 'zero' forem localizados em um limite de bytes pares... É possível obter dois bytes 'zero' em um limite ímpar quando cada parte de caracteres válidos. Por exemplo, x '01' x '00 x' 00 'x' 30 ' representa dois caracteres Unicode válidos e não finaliza a sequência como nula.

### ***VSCCSID (MQLONG) para MQCHARV***

Este é o identificador do conjunto de caracteres da sequência de comprimento variável endereçada pelo campo **VSPtr** ou **VSOffset** ..

O valor inicial desse campo é *MQCCSI\_APPL*, que é definido pelo MQ para indicar que ele deve ser alterado para o identificador do conjunto de caracteres verdadeiro do processo atual... Como resultado, o valor da constante *MQCCSI\_APPL* nunca é associado a uma sequência de comprimento variável..

O valor inicial desse campo pode ser alterado definindo um valor diferente para a constante *MQCCSI\_APPL* para sua unidade de compilação. Como você faz isso depende da linguagem de programação de seu aplicativo.

 Em sistemas z/OS , o aplicativo padrão CCSID usado por *MQCCSI\_APPL* é definido conforme a seguir:

- Para aplicativos LE em lote usando a interface DLL, o padrão é o CODESET associado ao código de idioma atual no momento em que **MQCONN** é emitido (o valor padrão é 1047).
- Para aplicativos LE em lote ligados a um dos stubs MQ em lote, o padrão é o CODESET associado ao código de idioma atual no momento da primeira chamada MQI emitida após **MQCONN** (o valor padrão é 1047).

- Para aplicativos não LE em lote em execução em um encadeamento z/OS UNIX System Services , o padrão é o valor de THLICCSID no momento da primeira chamada MQI emitida após **MQCONN** (o valor padrão é 1047).
- Para outros aplicativos em lote, o padrão é CCSID do gerenciador de fila.

## Redefinição de MQCCSI\_APPL

Os exemplos a seguir mostram como é possível substituir o valor de MQCCSI\_APPL em várias linguagens de programação. É possível alterar o valor de MQCCSI\_APPL, removendo a necessidade de configurar o VSCCSID para cada sequência de comprimento variável separadamente. Nesses exemplos, o CCSID é configurado como 1208; altere isso para o valor necessário. Esse se torna o valor padrão, que pode ser substituído configurando o VSCCSID em qualquer instância específica de MQCHARV..

Uso de C.

```
#define MQCCSI_APPL 1208
#include <cmqc.h>
```

Uso do COBOL

```
COPY CMQXYZV REPLACING -3 BY 1208.
```

uso de PL/I

```
%MQCCSI_APPL = '1208';
%include syslib(cmqp);
```

Uso do High Level Assembler

```
MQCCSI_APPL EQU 1208
CMQA LIST=NO
```

## MQCIH-cabeçalho CICS bridge





A estrutura MQCIH descreve as informações do cabeçalho para uma mensagem enviada para CICS no CICS bridge.

Para qualquer plataforma suportada pelo IBM MQ , é possível criar e transmitir uma mensagem que inclui a estrutura MQCIH, mas apenas um gerenciador de filas do IBM MQ for z/OS pode usar o CICS bridge. Portanto, para que a mensagem chegue ao CICS a partir de um gerenciador de filas nãoz/OS , sua rede do gerenciador de filas deve incluir pelo menos um gerenciador de filas do z/OS por meio do qual a mensagem pode ser roteada.

Todas as versões do CICS suportadas pelo IBM MQ 9.0.0e mais recente usam a versão fornecida pelo CICS da ponte. Para obter mais informações sobre como configurar o adaptador do IBM MQ CICS e os componentes do IBM MQ CICS bridge, veja a seção [Configurando conexões com o MQ](#) da documentação do CICS.

## Disponibilidade

A estrutura MQCIH está disponível nas plataformas a seguir:

-  AIX
-  Linux
-  Windows
-  z/OS

e para o IBM MQ MQI clients conectado a esses sistemas

## Nome do Formato

MQFMT\_CICS

## Versão

A versão atual de MQCIH é MQCIH\_VERSION\_2. Os campos que existem apenas na versão mais recente da estrutura são identificados como tal nas descrições a seguir:

Os arquivos de cabeçalho, COPY e INCLUDE fornecidos para as linguagens de programação suportadas contêm a versão mais recente de MQCIH, com o valor inicial do campo *Version* configurado como MQCIH\_VERSION\_2

## Conjunto de caracteres e codificação

Condições especiais se aplicam ao conjunto de caracteres e à codificação usados para a estrutura MQCIH e dados da mensagem do aplicativo:

- Os aplicativos que se conectarem ao gerenciador de filas que possui a fila do CICS bridge devem fornecer uma estrutura MQCIH que esteja no conjunto de caracteres e na codificação do gerenciador de fila. Isso ocorre porque a conversão de dados da estrutura MQCIH não é executada neste caso.
- Os aplicativos que se conectam a outros gerenciadores de filas podem fornecer uma estrutura MQCIH que esteja em qualquer um dos conjuntos de caracteres e codificações suportados; o agente do canal de mensagens de recebimento conectado ao gerenciador de filas que possui a fila CICS bridge converte a estrutura MQCIH.
- Os dados da mensagem do aplicativo após a estrutura MQCIH devem estar no mesmo conjunto de caracteres e na codificação que a estrutura MQCIH. Não é possível usar os campos *CodedCharSetId* e *Encoding* na estrutura MQCIH para especificar o conjunto de caracteres e a codificação dos dados da mensagem do aplicativo.

Deve-se fornecer uma saída de conversão de dados para converter os dados da mensagem do aplicativo se os dados não forem um dos formatos integrados suportados pelo gerenciador de filas.

## Uso

Se o aplicativo requerer valores que sejam iguais aos valores iniciais mostrados em [Tabela 477 na página 307](#) e a ponte estiver em execução com AUTH=LOCAL ou AUTH=IDENTIFY, será possível omitir a estrutura MQCIH da mensagem. Em todos os outros casos, a estrutura deve estar presente.

A ponte aceita uma estrutura MQCIH version-1 ou version-2, mas para transações 3270, você deve usar uma estrutura version-2.

O aplicativo deve assegurar que os campos documentados como campos de solicitação tenham valores apropriados na mensagem enviada para a ponte; esses campos são inseridos na ponte.

Os campos documentados como campos de resposta são configurados pelo CICS bridge na mensagem de resposta que a ponte envia para o aplicativo. As informações de erro são retornadas nos campos *ReturnCode*, *Function*, *CompCode*, *Reason* e *AbendCode*, mas nem todos eles são configurados em todos os casos. A tabela a seguir mostra quais campos são configurados para diferentes valores de *ReturnCode*.

<i>ReturnCode</i>	<i>Function</i>	<i>CompCode</i>	<i>Reason</i>	<i>AbendCode</i>
MQCRC_OK	-	-	-	-
MQCRC_BRIDGE_ERROR	-	-	MQFB_CICS_*	-

Tabela 476. Conteúdo de campos de informações de erro na estrutura MQCIH para MQCIH.. (continuação)

<b>ReturnCode</b>	<b>Function</b>	<b>CompCode</b>	<b>Reason</b>	<b>AbendCode</b>
MQCRC_MQ_API_ERROR MQCRC_BRIDGE_TIMEOUT	Nome da chamada do MQ	MQ CompCode	MQ Reason	-
MQCRC_CICS_EXEC_ERROR MQCRC_SECURITY_ERROR MQCRC_PROGRAM_NOT_AVAILABLE MQCRC_TRANSID_NOT_AVAILABLE	CICS EIBFN	CICS EIBRESP	CICS EIBRESP2	-
MQCRC_BRIDGE_ABEND MQCRC_APPLICATION_ABEND	-	-	-	CICS ABCODE

## Campos

**Nota:** Na tabela a seguir, os campos são agrupados por uso e não alfabeticamente. Os tópicos filhos seguem a mesma sequência.

<b>Nome e descrição do campo</b>	<b>Nome da constante</b>	<b>Valor inicial (se houver) da constante</b>
<u>StrucId</u> (identificador de estrutura)	MQCIH_STRUC_ID	' CIH '
<u>Versão</u> (número da versão da estrutura).	MQCIH_VERSION_2	2
<u>StrucLength</u> (comprimento da estrutura MQCIH)	MQCIH_LENGTH_2	180
<u>Codificação</u> (reservado)	Nenhum	0
<u>CodedCharSetId</u> (reservado)	Nenhum	0
<u>Formato</u> (nome do formatoMQ de dados que seguem MQCIH)	MQFMT_NONE	Espaços em branco
<u>Sinalizadores</u> (sinalizadores)	MQCIH_NONE	0
<u>ReturnCode</u> (código de retorno da ponte)	MQCRC_OK	0
<u>CompCode</u> (código de conclusãoMQ ou CICS EIBRESP)	MQCC_OK	0
<u>Razão</u> (código de razão ou feedback doMQ ou CICS EIBRESP2)	MQRC_NONE	0
<u>UOWControl</u> (controle de unidade de trabalho).	MQCUOWC_ONLY	273
<u>GetWaitIntervalo</u> (intervalo de espera para chamada MQGET emitida pela tarefa de ponte)	MQCGWI_DEFAULT	-2
<u>LinkType</u> (tipo de link)	MQCLT_PROGRAM	1
<u>OutputDataComprimento</u> (comprimento de dados COMMAREA de saída)	MQCODL_AS_INPUT	-1
<u>FacilityKeepTime</u> (tempo de liberação do recurso de Ponte).	Nenhum	0
<u>ADSDescriptor</u> (enviar / receber descritor ADS)	MQCADSD_NONE	0
<u>ConversationalTask</u> (se a tarefa pode ser conversacional)	MQCCT_NO	0

Tabela 477. Campos em MQCIH para MQCIH. (continuação)

Nome e descrição do campo	Nome da constante	Valor inicial (se houver) da constante
<u>TaskEnd</u> (status no final da tarefa)	MQCTES_NOSYNC	0
<u>Recurso</u> (token de recurso de ponte).	MQCFAC_NONE	Nulos
<u>Função</u> (nome da chamadaMQ ou CICS função EIBFN)	MQCFUNC_NONE	Espaços em branco
<u>AbendCode</u> (código de encerramento anormal)	Nenhum	Espaços em branco
<u>Autenticador</u> (senha ou passticket)	Nenhum	Espaços em branco
<u>Reserved1</u> (reservado)	Nenhum	Espaços em branco
<u>ReplyToFormat</u> (nome do formato da mensagem de resposta doMQ )	MQFMT_NONE	Espaços em branco
<u>RemoteSysId</u> (ID do sistema remoto CICS a ser usado)	Nenhum	Espaços em branco
<u>RemoteTransId</u> (CICS RTRANSID a ser usado)	Nenhum	Espaços em branco
<u>TransactionId</u> (transação para anexar)	Nenhum	Espaços em branco
<u>FacilityLike</u> (atributos emulados do terminal);	Nenhum	Espaços em branco
<u>AttentionId</u> (chave AID)	Nenhum	Espaços em branco
<u>StartCode</u> (código de início da transação)	MQCSC_NONE	Espaços em branco
<u>CancelCode</u> (código de transação de encerramento anormal)	Nenhum	Espaços em branco
<u>NextTransactionId</u> (próxima transação a ser anexada)	Nenhum	Espaços em branco
<u>Reserved2</u> (reservado)	Nenhum	Espaços em branco
<u>Reserved3</u> (reservado)	Nenhum	Espaços em branco
<b>Nota:</b> Os campos restantes não estarão presentes se <i>Version</i> for menor que MQCIH_VERSION_2.		
<u>CursorPosition</u> (posição do cursor)	Nenhum	0
<u>ErrorOffset</u> (compensação de erro na mensagem)	Nenhum	0
<u>InputItem</u> (item de entrada)	Nenhum	0
<u>Reserved4</u> (reservado)	Nenhum	0
<p><b>Notas:</b></p> <ol style="list-style-type: none"> <li>1. O símbolo – representa um único caractere em branco.</li> <li>2. Na linguagem de programação C, a variável macroMQCIH_DEFAULT contém os valores que estão listados na tabela. Use-o da seguinte maneira para fornecer valores iniciais para os campos na estrutura:</li> </ol> <pre style="background-color: #f0f0f0; padding: 5px;">MQCIH MyCIH = {MQCIH_DEFAULT};</pre>		



## Declarações de idiomas

### Declaração C para MQCIH

```
typedef struct tagMQCIH MQCIH;
struct tagMQCIH {
    MQCHAR4  StrucId;           /* Structure identifier */
    MQLONG   Version;          /* Structure version number */
    MQLONG   StrucLength;      /* Length of MQCIH structure */
    MQLONG   Encoding;         /* Reserved */
    MQLONG   CodedCharSetId;   /* Reserved */
    MQCHAR8  Format;           /* MQ format name of data that follows
                               MQCIH */
    MQLONG   Flags;            /* Flags */
    MQLONG   ReturnCode;       /* Return code from bridge */
    MQLONG   CompCode;         /* MQ completion code or CICS EIBRESP */
    MQLONG   Reason;          /* MQ reason or feedback code, or CICS
                               EIBRESP2 */
    MQLONG   UOWControl;       /* Unit-of-work control */
    MQLONG   GetWaitInterval;  /* Wait interval for MQGET call issued
                               by bridge task */
    MQLONG   LinkType;         /* Link type */
    MQLONG   OutputDataLength; /* Output COMMAREA data length */
    MQLONG   FacilityKeepTime; /* Bridge facility release time */
    MQLONG   ADSDescriptor;    /* Send/receive ADS descriptor */
    MQLONG   ConversationalTask; /* Whether task can be conversational */
    MQLONG   TaskEndStatus;    /* Status at end of task */
    MQBYTE8  Facility;         /* Bridge facility token */
    MQCHAR4  Function;         /* MQ call name or CICS EIBFN
                               function */
    MQCHAR4  AbendCode;        /* Abend code */
    MQCHAR8  Authenticator;    /* Password or passticket */
    MQCHAR8  Reserved1;        /* Reserved */
    MQCHAR8  ReplyToFormat;    /* MQ format name of reply message */
    MQCHAR4  RemoteSysId;      /* Reserved */
    MQCHAR4  RemoteTransId;    /* Reserved */
    MQCHAR4  TransactionId;    /* Transaction to attach */
    MQCHAR4  FacilityLike;     /* Terminal emulated attributes */
    MQCHAR4  AttentionId;      /* AID key */
    MQCHAR4  StartCode;        /* Transaction start code */
    MQCHAR4  CancelCode;       /* Abend transaction code */
    MQCHAR4  NextTransactionId; /* Next transaction to attach */
    MQCHAR8  Reserved2;        /* Reserved */
    MQCHAR8  Reserved3;        /* Reserved */
    MQLONG   CursorPosition;   /* Cursor position */
    MQLONG   ErrorOffset;      /* Offset of error in message */
    MQLONG   InputItem;        /* Reserved */
    MQLONG   Reserved4;        /* Reserved */
};
```

### Declaração COBOL para MQCIH

```
** MQCIH structure
10 MQCIH.
** Structure identifier
15 MQCIH-STRUCID PIC X(4).
** Structure version number
15 MQCIH-VERSION PIC S9(9) BINARY.
** Length of MQCIH structure
15 MQCIH-STRUCLENGTH PIC S9(9) BINARY.
** Reserved
15 MQCIH-ENCODING PIC S9(9) BINARY.
** Reserved
15 MQCIH-CODEDCHARSETID PIC S9(9) BINARY.
** MQ format name of data that follows MQCIH
15 MQCIH-FORMAT PIC X(8).
** Flags
15 MQCIH-FLAGS PIC S9(9) BINARY.
** Return code from bridge
15 MQCIH-RETURNCODE PIC S9(9) BINARY.
** MQ completion code or CICS EIBRESP
15 MQCIH-COMPCODE PIC S9(9) BINARY.
** MQ reason or feedback code, or CICS EIBRESP2
15 MQCIH-REASON PIC S9(9) BINARY.
** Unit-of-work control
15 MQCIH-UOWCONTROL PIC S9(9) BINARY.
** Wait interval for MQGET call issued by bridge task
```

```

15 MQCIH-GETWAITINTERVAL PIC S9(9) BINARY.
** Link type
15 MQCIH-LINKTYPE PIC S9(9) BINARY.
** Output COMMAREA data length
15 MQCIH-OUTPUTDATALENGTH PIC S9(9) BINARY.
** Bridge facility release time
15 MQCIH-FACILITYKEEPTIME PIC S9(9) BINARY.
** Send/receive ADS descriptor
15 MQCIH-ADSDESCRIPTOR PIC S9(9) BINARY.
** Whether task can be conversational
15 MQCIH-CONVERSATIONALTASK PIC S9(9) BINARY.
** Status at end of task
15 MQCIH-TASKENDSTATUS PIC S9(9) BINARY.
** Bridge facility token
15 MQCIH-FACILITY PIC X(8).
** MQ call name or CICS EIBFN function
15 MQCIH-FUNCTION PIC X(4).
** Abend code
15 MQCIH-ABENDCODE PIC X(4).
** Password or passticket
15 MQCIH-AUTHENTICATOR PIC X(8).
** Reserved
15 MQCIH-RESERVED1 PIC X(8).
** MQ format name of reply message
15 MQCIH-REPLYTOFORMAT PIC X(8).
** Reserved
15 MQCIH-REMOTESYSID PIC X(4).
** Reserved
15 MQCIH-REMOTETRANSID PIC X(4).
** Transaction to attach
15 MQCIH-TRANSACTIONID PIC X(4).
** Terminal emulated attributes
15 MQCIH-FACILITYLIKE PIC X(4).
** AID key
15 MQCIH-ATTENTIONID PIC X(4).
** Transaction start code
15 MQCIH-STARTCODE PIC X(4).
** Abend transaction code
15 MQCIH-CANCELCODE PIC X(4).
** Next transaction to attach
15 MQCIH-NEXTTRANSACTIONID PIC X(4).
** Reserved
15 MQCIH-RESERVED2 PIC X(8).
** Reserved
15 MQCIH-RESERVED3 PIC X(8).
** Cursor position
15 MQCIH-CURSORPOSITION PIC S9(9) BINARY.
** Offset of error in message
15 MQCIH-ERROROFFSET PIC S9(9) BINARY.
** Reserved
15 MQCIH-INPUTITEM PIC S9(9) BINARY.
** Reserved
15 MQCIH-RESERVED4 PIC S9(9) BINARY.

```

## Declaração PL/I para MQCIH

```

dcl
1 MQCIH based,
3 StrucId char(4), /* Structure identifier */
3 Version fixed bin(31), /* Structure version number */
3 StrucLength fixed bin(31), /* Length of MQCIH structure */
3 Encoding fixed bin(31), /* Reserved */
3 CodedCharSetId fixed bin(31), /* Reserved */
3 Format char(8), /* MQ format name of data that
follows MQCIH */
3 Flags fixed bin(31), /* Flags */
3 ReturnCode fixed bin(31), /* Return code from bridge */
3 CompCode fixed bin(31), /* MQ completion code or CICS
EIBRESP */
3 Reason fixed bin(31), /* MQ reason or feedback code, or
CICS EIBRESP2 */
3 UOWControl fixed bin(31), /* Unit-of-work control */
3 GetWaitInterval fixed bin(31), /* Wait interval for MQGET call
issued by bridge task */
3 LinkType fixed bin(31), /* Link type */
3 OutputDataLength fixed bin(31), /* Output COMMAREA data length */
3 FacilityKeepTime fixed bin(31), /* Bridge facility release time */
3 ADSDescriptor fixed bin(31), /* Send/receive ADS descriptor */
3 ConversationalTask fixed bin(31), /* Whether task can be

```

```

3 TaskEndStatus      fixed bin(31), /* conversational */
3 Facility           char(8), /* Status at end of task */
3 Function           char(4), /* Bridge facility token */
                    /* MQ call name or CICS EIBFN
                    function */
3 AbendCode         char(4), /* Abend code */
3 Authenticator      char(8), /* Password or passticket */
3 Reserved1         char(8), /* Reserved */
3 ReplyToFormat     char(8), /* MQ format name of reply
                    message */
3 RemoteSysId       char(4), /* Reserved */
3 RemoteTransId     char(4), /* Reserved */
3 TransactionId     char(4), /* Transaction to attach */
3 FacilityLike      char(4), /* Terminal emulated attributes */
3 AttentionId       char(4), /* AID key */
3 StartCode         char(4), /* Transaction start code */
3 CancelCode        char(4), /* Abend transaction code */
3 NextTransactionId char(4), /* Next transaction to attach */
3 Reserved2         char(8), /* Reserved */
3 Reserved3         char(8), /* Reserved */
3 CursorPosition    fixed bin(31), /* Cursor position */
3 ErrorOffset       fixed bin(31), /* Offset of error in message */
3 InputItem         fixed bin(31), /* Reserved */
3 Reserved4         fixed bin(31); /* Reserved */

```

### Declaração High Level Assembler para MQCIH

```

MQCIH                DSECT
MQCIH_STRUCID        DS  CL4  Structure identifier
MQCIH_VERSION        DS  F    Structure version number
MQCIH_STRUCLNGTH     DS  F    Length of MQCIH structure
MQCIH_ENCODING       DS  F    Reserved
MQCIH_CODEDCHARSETID DS  F    Reserved
MQCIH_FORMAT         DS  CL8  MQ format name of data that follows
*
MQCIH_FLAGS          DS  F    Flags
MQCIH_RETURNCODE     DS  F    Return code from bridge
MQCIH_COMPCODE       DS  F    MQ completion code or CICS EIBRESP
MQCIH_REASON         DS  F    MQ reason or feedback code, or CICS
*
MQCIH_UOWCONTROL     DS  F    Unit-of-work control
MQCIH_GETWAITINTERVAL DS  F    Wait interval for MQGET call issued
*
MQCIH_LINKTYPE       DS  F    Link type
MQCIH_OUTPUTDATALENGTH DS  F    Output COMMAREA data length
MQCIH_FACILITYKEEPTIME DS  F    Bridge facility release time
MQCIH_ADSDESCRIPTOR  DS  F    Send/receive ADS descriptor
MQCIH_CONVERSATIONALTASK DS  F    Whether task can be conversational
MQCIH_TASKENDSTATUS  DS  F    Status at end of task
MQCIH_FACILITY       DS  XL8  Bridge facility token
MQCIH_FUNCTION        DS  CL4  MQ call name or CICS EIBFN function
MQCIH_ABENDCODE      DS  CL4  Abend code
MQCIH_AUTHENTICATOR  DS  CL8  Password or passticket
MQCIH_RESERVED1     DS  CL8  Reserved
MQCIH_REPLYTOFORMAT  DS  CL8  MQ format name of reply message
MQCIH_REMOTESYSID    DS  CL4  Reserved
MQCIH_REMOTETRANSID  DS  CL4  Reserved
MQCIH_TRANSACTIONID  DS  CL4  Transaction to attach
MQCIH_FACILITYLIKE   DS  CL4  Terminal emulated attributes
MQCIH_ATTENTIONID    DS  CL4  AID key
MQCIH_STARTCODE      DS  CL4  Transaction start code
MQCIH_CANCELCODE     DS  CL4  Abend transaction code
MQCIH_NEXTTRANSACTIONID DS  CL4  Next transaction to attach
MQCIH_RESERVED2     DS  CL8  Reserved
MQCIH_RESERVED3     DS  CL8  Reserved
MQCIH_CURSORPOSITION DS  F    Cursor position
MQCIH_ERROROFFSET    DS  F    Offset of error in message
MQCIH_INPUTITEM      DS  F    Reserved
MQCIH_RESERVED4     DS  F    Reserved
*
MQCIH_LENGTH         EQU  *-MQCIH
MQCIH_AREA           ORG  MQCIH
                    DS  CL(MQCIH_LENGTH)

```

### Declaração do Visual Basic para MQCIH.

```
Type MQCIH
```

StrucId	As String*4	'Structure identifier'
Version	As Long	'Structure version number'
StrucLength	As Long	'Length of MQCIH structure'
Encoding	As Long	'Reserved'
CodedCharSetId	As Long	'Reserved'
Format	As String*8	'MQ format name of data that follows' 'MQCIH'
Flags	As Long	'Flags'
ReturnCode	As Long	'Return code from bridge'
CompCode	As Long	'MQ completion code or CICS EIBRESP'
Reason	As Long	'MQ reason or feedback code, or CICS' 'EIBRESP2'
UOWControl	As Long	'Unit-of-work control'
GetWaitInterval	As Long	'Wait interval for MQGET call issued' 'by bridge task'
LinkType	As Long	'Link type'
OutputDataLength	As Long	'Output COMMAREA data length'
FacilityKeepTime	As Long	'Bridge facility release time'
ADSDescriptor	As Long	'Send/receive ADS descriptor'
ConversationalTask	As Long	'Whether task can be conversational'
TaskEndStatus	As Long	'Status at end of task'
Facility	As MQBYTE8	'Bridge facility token'
Function	As String*4	'MQ call name or CICS EIBFN function'
AbendCode	As String*4	'Abend code'
Authenticator	As String*8	'Password or passticket'
Reserved1	As String*8	'Reserved'
ReplyToFormat	As String*8	'MQ format name of reply message'
RemoteSysId	As String*4	'Reserved'
RemoteTransId	As String*4	'Reserved'
TransactionId	As String*4	'Transaction to attach'
FacilityLike	As String*4	'Terminal emulated attributes'
AttentionId	As String*4	'AID key'
StartCode	As String*4	'Transaction start code'
CancelCode	As String*4	'Abend transaction code'
NextTransactionId	As String*4	'Next transaction to attach'
Reserved2	As String*8	'Reserved'
Reserved3	As String*8	'Reserved'
CursorPosition	As Long	'Cursor position'
ErrorOffset	As Long	'Offset of error in message'
InputItem	As Long	'Reserved'
Reserved4	As Long	'Reserved'
End Type		

### **StrucId (MQCHAR4) para MQCIH**

Este é o identificador de estrutura da estrutura do cabeçalho de informações CICS . Ele é sempre um campo de entrada Seu valor é MQCIH\_STRUC\_ID.

O valor deve ser:.

#### **MQCIH\_STRUC\_ID**

Identificador para a estrutura do cabeçalho de informações CICS .

Para a linguagem de programação C, a constante MQCIH\_STRUC\_ID\_ARRAY também é definida. Ele tem o mesmo valor que MQCIH\_STRUC\_ID, mas é uma matriz de caracteres em vez de uma sequência.

### **Versão (MQLONG) para MQCIH**

Este campo é um campo de solicitação Seu valor inicial é MQCIH\_VERSION\_2.

O valor deve ser um dos seguintes:

#### **MQCIH\_VERSION\_1**

Version-1 CICS estrutura do cabeçalho de informações.

#### **MQCIH\_VERSION\_2**

Version-2 CICS estrutura do cabeçalho de informações.

Os campos que existem apenas na versão mais recente da estrutura são identificados como tal nas descrições dos campos A constante a seguir especifica o número da versão atual:

#### **MQCIH\_CURRENT\_VERSION**

Versão atual da estrutura do cabeçalho de informações CICS .

### **StrucLength (MQLONG) para MQCIH**

Esse é um campo de solicitação, com um valor inicial de MQCIH\_LENGTH\_2.

O valor deve ser um dos seguintes:

#### **MQCIH\_LENGTH\_1**

Comprimento da estrutura do cabeçalho de informações version-1 CICS .

#### **MQCIH\_LENGTH\_2**

Comprimento da estrutura do cabeçalho de informações version-2 CICS .

A constante a seguir especifica o comprimento da versão atual:

#### **MQCIH\_CURRENT\_LENGTH**

Comprimento da versão atual da estrutura do cabeçalho de informações do CICS

### **Codificação (MQLONG) para MQCIH**

Este campo é um campo reservado; seu valor não é significativo. Seu valor inicial é 0.

A Codificação para estruturas suportadas que seguem uma estrutura MQCIH é igual à Codificação da própria estrutura MQCIH e obtida de qualquer cabeçalho IBM MQ anterior.

### **CodedCharSetId (MQLONG) para MQCIH**

CodedCharSetId é um campo reservado; seu valor não é significativo. O valor inicial desse campo é 0.

O ID do Conjunto de Caracteres para estruturas suportadas que seguem uma estrutura MQCIH é o mesmo que o ID do Conjunto de Caracteres da própria estrutura MQCIH e é obtido de qualquer cabeçalho IBM MQ anterior.

### **Formato (MQCHAR8) para MQCIH**

Este campo mostra o nome do formato IBM MQ dos dados que seguem a estrutura MQCIH..

Na chamada MQPUT ou MQPUT1 , o aplicativo deve configurar esse campo para o valor apropriado aos dados. As regras para codificar esse campo são as mesmas regras para codificar o campo *Format* no MQMD.

Esse nome de formato também será usado na mensagem de resposta, se o campo *ReplyToFormat* tiver o valor MQFMT\_NONE.

- Para solicitações de DPL, *Format* deve ser o nome do formato da COMMAREA
- Para solicitações 3270, o *Format* deve ser CSQCBDCIe a ponte configura o formato como CSQCBDCO para mensagens de Resposta

As saídas de conversão de dados para esses formatos devem ser instaladas no gerenciador de filas no qual elas devem ser executadas

Se a mensagem de solicitação gerar uma mensagem de resposta de erro, a mensagem de resposta de erro terá um nome de formato de MQFMT\_STRING

Este campo é um campo de solicitação O comprimento desse campo é fornecido por MQ\_FORMAT\_LENGTH O valor inicial desse campo é MQFMT\_NONE.

### **Sinalizadores (MQLONG) para MQCIH**

Este campo é um campo de solicitação O valor inicial deste campo é MQCIH\_NONE.

O valor deve ser::

#### **MQCIH\_NONE**

Sem bandeiras.

#### **MQCIH\_PASS\_EXPIRATION**

A mensagem de resposta contém:

- As mesmas opções de relatório de expiração que a mensagem de solicitação

- O tempo de expiração restante da mensagem de solicitação sem ajuste feito para o tempo de processamento da ponte..

Se você omitir esse valor, o tempo de expiração será configurado como *unlimited*.

#### **MQCIH\_REPLY\_WITHOUT\_NULLS**

O comprimento da mensagem de resposta de um pedido do programa DPL CICS é ajustada para excluir nulos finais (X'00 ') no final do COMMAREA retornado pelo programa DPL. Se esse valor não for configurado, os nulos poderão ser significativos e o COMMAREA completo será retornado.

#### **MQCIH\_SYNC\_ON\_RETURN**

O link CICS para solicitações de DPL usa a opção SYNCONRETURN, fazendo com que o CICS tome um ponto de sincronização quando o programa for concluído se ele for enviado para outra região CICS . A ponte não especifica para qual região do CICS enviar a solicitação; isso é controlado pela definição do programa CICS ou recursos de balanceamento de carga de trabalho..

#### **ReturnCode (MQLONG) para MQCIH**

O valor desse campo é o código de retorno do CICS bridge que descreve o resultado do processamento executado pela ponte... Esse campo é um campo de resposta, com um valor inicial de MQCRC\_OK.

Os campos *Function*, *CompCode*, *Reason* e *AbendCode* podem conter informações adicionais (consulte Tabela 476 na página 306 ) O valor é um dos seguintes:

#### **MQCRC\_APPLICATION\_ABEND**

(5, X'005 ') O aplicativo terminou de forma anormal.

#### **MQCRC\_BRIDGE\_ABEND**

(4, X'004 ') CICS bridge terminou de forma anormal.

#### **MQCRC\_BRIDGE\_ERROR**

(3, X'003 ') CICS bridge detectou um erro.

#### **MQCRC\_BRIDGE\_TIMEOUT**

(8, X'008 ') Segunda ou posterior mensagem na unidade de trabalho atual não recebida dentro do tempo especificado.

#### **MQCRC\_CICS\_EXEC\_ERROR**

(1, X'001 ') instrução EXEC CICS detectou um erro.

#### **MQCRC\_MQ\_API\_ERROR**

(2, X'002 ') chamada MQ detectou um erro.

#### **MQCRC\_OK**

(0, X'000 ') Nenhum erro.

#### **MQCRC\_PROGRAM\_NOT\_AVAILABLE**

(7, X'007 ') Programa não disponível.

#### **MQCRC\_SECURITY\_ERROR**

(6, X'006 ') Ocorreu um erro de segurança

#### **MQCRC\_TRANSID\_NOT\_AVAILABLE**

(9, X'009 ') Transação não disponível.

#### **CompCode (MQLONG) para MQCIH**

Este campo é um campo de resposta Seu valor inicial é MQCC\_OK

O valor retornado nesse campo depende de *ReturnCode* ; Consulte [Tabela 476 na página 306](#).

#### **Razão (MQLONG) para MQCIH**

Este campo é um campo de resposta Seu valor inicial é MQRC\_NONE.

O valor retornado nesse campo depende de *ReturnCode* ; Consulte [Tabela 476 na página 306](#).

#### **UOWControl (MQLONG) para MQCIH**

Esse campo é um campo de solicitação que controla o processamento da unidade de trabalho executado pelo CICS bridge O valor inicial desse campo é MQCUOWC\_ONLY.

É possível solicitar que a ponte execute uma única transação ou um ou mais programas em uma unidade de trabalho. O campo indica se o CICS bridge inicia uma unidade de serviço, executa a função solicitada dentro da unidade de trabalho atual ou termina a unidade de trabalho confirmando-a ou fazendo backup dela. Várias combinações são suportadas, para otimizar os fluxos de transmissão de dados

O valor deve ser um dos seguintes:

**MQCUOWC\_ONLY**

Iniciar unidade de trabalho, executar função e, em seguida, confirmar a unidade de trabalho.

**MQCUOWC\_CONTINUE**

Dados adicionais para a unidade de trabalho atual (3270 apenas)

**MQCUOWC\_FIRST**

Iniciar unidade de trabalho e executar função.

**MQCUOWC\_MIDDLE**

Executar função dentro da unidade de trabalho atual

**MQCUOWC\_LAST**

Execute a função e, em seguida, confirme a unidade de trabalho

**MQCUOWC\_COMMIT**

Confirmar a unidade de trabalho (somente DPL).

**MQCUOWC\_BACKOUT**

Voltar a unidade de trabalho (somente DPL).

***Intervalo de GetWait(MQLONG) para MQCIH***

Este campo é um campo de solicitação Seu valor inicial é MQCGWI\_DEFAULT.

Esse campo se aplica apenas quando *UOWControl* tem o valor MQCUOWC\_FIRST. Ele permite que o aplicativo de envio especifique o tempo aproximado em milissegundos que as chamadas MQGET emitidas pela ponte esperarão pela segunda e pelas mensagens de solicitação subsequentes para a unidade de trabalho iniciada por essa mensagem. Esse recurso substitui o intervalo de espera padrão usado pela ponte. É possível usar os seguintes valores especiais:

**MQCGWI\_DEFAULT**

Intervalo de espera padrão.

Esse valor faz o CICS bridge aguardar o tempo especificado quando a ponte foi iniciada.

**MQWI\_UNLIMITED**

Intervalo de espera ilimitado.

***LinkType (MQLONG) para MQCIH***

Este campo é um campo de solicitação Seu valor inicial é MQCLT\_PROGRAM.

Esse valor indica o tipo de objeto que a bridge tenta vincular. Deve ser um dos valores a seguir:

**MQCLT\_PROGRAM**

Programa DPL.

**MQCLT\_TRANSACTION**

Transação 3270..

***Comprimento de OutputData(MQLONG) para MQCIH***

Este campo é um campo de pedido utilizado apenas para programas DPL Seu valor inicial é MQCODL\_AS\_INPUT.

Esse valor é o comprimento de dados do usuário a serem retornados para o cliente em uma mensagem de resposta Esse comprimento inclui o nome do programa de 8 bytes O comprimento do COMMAREA passado para o programa vinculado é o máximo deste campo e o comprimento dos dados do usuário na mensagem de solicitação, menos 8.

**Nota:** O comprimento dos dados do usuário em uma mensagem é o comprimento da mensagem excluindo a estrutura MQCIH.

Se o comprimento dos dados do usuário na mensagem de solicitação for menor que *OutputDataLength*, a opção *DATALENGTH* do comando *LINK* será usada, permitindo que o *LINK* seja enviado de forma eficiente para outra região *CICS*.

É possível usar o seguinte valor especial:

#### **MQCODL\_AS\_INPUT**

O comprimento de saída é igual ao comprimento de entrada.

Esse valor pode ser necessário mesmo se nenhuma resposta for solicitada, para assegurar que a *COMMAREA* transmitida para o programa vinculado seja de tamanho suficiente.

#### **FacilityKeepTime (MQLONG) para MQCIH**

*FacilityKeepTime* é o período de tempo em segundos que o recurso de ponte é mantido após o término da transação do usuário.

Para transações pseudo-conversacionais, especifique um valor que corresponda à duração esperada de uma pseudo-conversa o; especifique zero para a  ltima transa o de uma pseudo-conversa o e, para outros tipos de transa o, especifique zero.

Este campo   um campo de pedido utilizado apenas para transa o 3270 O valor inicial desse campo   0.

#### **ADSDescriptor (MQLONG) para MQCIH**

Este campo   um indicador que especifica se deve enviar descritores ADS em solicita o *SEND* e *RECEIVE BMS*.

Os seguintes valores s o definidos:

#### **MQCADSD\_NONE**

N o enviar ou receber descritores ADS.

#### **MQCADSD\_SEND**

Enviar descritores do ADS

#### **MQCADSD\_RECV**

Receber descritores ADS.

#### **MQCADSD\_MSGFORMAT**

Use o formato da mensagem para os descritores do ADS

Isso envia ou recebe os descritores ADS usando a forma longa do descritor ADS. O formul rio longo possui campos alinhados em limites de 4 bytes.

Configure o campo *ADSDescriptor* da seguinte forma:

- Se n o estiver usando descritores ADS, configure o campo para *MQCADSD\_NONE*.
- Se voc  estiver usando descritores ADS com o *mesmo* *CCSID* em cada ambiente, configure o campo para a soma de *MQCADSD\_SEND* e *MQCADSD\_RECV*.
- Se estiver usando descritores ADS com *CCSIDs diferentes* em cada ambiente, configure o campo para a soma de *MQCADSD\_SEND*, *MQCADSD\_RECV* e *MQCADSD\_MSGFORMAT*.

Este   um campo de pedido utilizado apenas para transa o 3270 O valor inicial deste campo   *MQCADSD\_NONE*.

#### **ConversationalTask (MQLONG) para MQCIH**

Este campo   um indicador que especifica se deve permitir que a tarefa emita solicita o para obter mais informa o ou parar a tarefa e emitir uma mensagem de encerramento de forma anormal.

O valor deve ser uma das seguintes op oes:

#### **MQCCT\_YES**

A tarefa   de conversa o

#### **MQCCT\_NO**

A tarefa n o   de conversa o



Este campo é um campo de pedido utilizado apenas para transações 3270 O valor inicial desse campo é MQCCT\_NO.

### **TaskEndStatus (MQLONG) para MQCIH**

Este campo é um campo de resposta, mostrando o status da transação do usuário no término da tarefa O campo é usado apenas para transações 3270 e seu valor inicial é MQCTES\_NOSYNC.

Um dos seguintes valores é retornado:

#### **MQCTES\_NOSYNC**

Não sincronizado..

A transação do usuário ainda não foi concluída e não possui sincronização. Nesse caso, o campo *MsgType* no MQMD é MQMT\_REQUEST

#### **MQCTES\_COMMIT**

Confirmar unidade de trabalho.

A transação do usuário ainda não foi concluída, mas sincroniza a primeira unidade de trabalho. O campo *MsgType* em MQMD é MQMT\_DATAGRAM, neste caso,

#### **MQCTES\_BACKOUT**

Unidade de trabalho de volta.

A transação do usuário ainda não foi concluída A unidade de trabalho atual é restaurada. O campo *MsgType* em MQMD é MQMT\_DATAGRAM, neste caso,

#### **MQCTES\_ENDTASK**

Terminar tarefa.

A transação do usuário foi encerrada (ou encerrada de forma anormal). Neste caso, o campo *MsgType* em MQMD é MQMT\_REPLY

### **Recurso (MQBYTE8) para MQCIH**

Este campo mostra o token do recurso de ponte de 8 bytes

Um token de recurso de ponte permite que várias transações em uma pseudo-conversa usem o mesmo recurso de ponte (terminal 3270 virtual). Na primeira, ou única, mensagem em uma pseudo-conversa, configure um valor de MQCFAC\_NONE. Esse valor informa CICS para alocar um novo recurso de ponte para essa mensagem. Um token de recurso de Ponte é retornado em mensagens de resposta quando um *FacilityKeepTime* diferente de zero é especificado na mensagem de entrada. As mensagens de entrada subsequentes dentro de uma pseudoconversa devem, então, usar o mesmo token de recurso de ponte

O seguinte valor especial é definido:

#### **MQCFAC\_NONE**

Nenhum token de recurso especificado.

Para a linguagem de programação C, a constante MQCFAC\_NONE\_ARRAY também é definida e tem o mesmo valor que MQCFAC\_NONE, mas é uma matriz de caracteres em vez de uma sequência.

Esse campo é um campo de solicitação e de resposta utilizado apenas para transações 3270. O comprimento desse campo é fornecido por MQ\_FUNCTION\_LENGTH O valor inicial desse campo é MQCFAC\_NONE.

### **Função (MQCHAR4) para MQCIH**

Este campo é um campo de resposta O comprimento desse campo é fornecido por MQ\_FUNCTION\_LENGTH O valor inicial desse campo é MQCFUNC\_NONE.

O valor retornado nesse campo depende de *ReturnCode* ; Consulte [Tabela 476 na página 306](#). Os seguintes valores são possíveis quando *Function* contém um nome de chamada IBM MQ :

#### **MQCFUNC\_MQCONN**

Chamada de MQCONN

**MQCFUNC\_MQGET**

Chamada MQGET.

**MQCFUNC\_MQINQ**

Chamada de MQINQ

**MQCFUNC\_MQOPEN**

chamada MQOPEN.

**MQCFUNC\_MQPUT**

chamada MQPUT.

**MQCFUNC\_MQPUT1**

Chamada MQPUT1 .

**MQCFUNC\_NONE**

Nenhuma chamada.

Em todos os casos, para a linguagem de programação C, as constantes MQCFUNC\_\*\_ARRAY também são definidas; essas constantes têm os mesmos valores que as constantes MQCFUNC\_\* correspondentes, mas são matrizes de caracteres em vez de sequências.

***AbendCode (MQCHAR4) para MQCIH***

AbendCode é um campo de resposta.. O comprimento desse campo é fornecido por MQ\_ABEND\_CODE\_LENGTH O valor inicial deste campo é 4 caracteres em branco.

O valor retornado neste campo será significativo apenas se o campo *ReturnCode* tiver o valor MQCRC\_APPLICATION\_ABEND ou MQCRC\_BRIDGE\_ABEND. Se houver, *AbendCode* conterá o valor CICS ABCODE.

***Autenticador (MQCHAR8) para MQCIH***

O valor desse campo é a senha ou passticket.

Se a autenticação do identificador de usuário estiver ativa para o CICS bridge, *Authenticator* será usado com o identificador de usuário no contexto de identidade do MQMD para autenticar o emissor da mensagem.

Este é um campo de solicitação O comprimento desse campo é fornecido por MQ\_AUTHENTICATOR\_LENGTH O valor inicial deste campo é 8 espaços em branco.

***Reserved1 (MQCHAR8) para MQCIH***

Este campo é um campo reservado O valor deve ser 8 espaços em branco.

***ReplyTo(MQCHAR8) para MQCIH***

O valor desse campo é o nome do formato IBM MQ da mensagem de resposta enviada em resposta à mensagem atual.

As regras para codificar esse campo são as mesmas que aquelas regras para codificar o campo *Format* no MQMD

Este campo é um campo de pedido utilizado apenas para programas DPL O comprimento desse campo é fornecido por MQ\_FORMAT\_LENGTH O valor inicial desse campo é MQFMT\_NONE.

***RemoteSysId (MQCHAR4) para MQCIH***

Este campo mostra o identificador do sistema CICS do sistema CICS que está processando a solicitação.

Se este campo estiver em branco, a solicitação do sistema CICS será processada no mesmo sistema CICS que o monitor de ponte O SYSID usado é retornado na mensagem de Resposta..

Para uma pseudo-conversa 3270, todas as mensagens subsequentes na conversa devem especificar o SYSID remoto retornado na resposta inicial. Se especificado, o SYSID deverá:

- Esteja ativo

- Ter acesso à fila de Solicitações do IBM MQ
- Seja acessível pelos links do ISC do CICS a partir do sistema CICS do monitor de ponte

### ***RemoteTransId (MQCHAR4) para MQCIH***

Esse campo é um campo de Solicitação opcional O comprimento desse campo é fornecido por MQ\_TRANSACTION\_ID\_LENGTH

Se especificado, o campo será usado como o valor RTRANSID de CICS START.

### ***TransactionId (MQCHAR4) para MQCIH***

Este campo é um campo de solicitação Seu comprimento é fornecido por MQ\_TRANSACTION\_ID\_LENGTH O valor inicial deste campo é quatro espaços em branco.

Se *LinkType* tiver o valor MQCLT\_TRANSACTION, *TransactionId* será o identificador de transação da transação do usuário a ser executada; especifique um valor não em branco nesse caso.

Se *LinkType* tiver o valor MQCLT\_PROGRAM, *TransactionId* será o código de transação sob o qual todos os programas dentro da unidade de trabalho devem ser executados. Se você especificar um valor em branco, o CICS DPL bridge default transaction code (CKBP) será usado. Se o valor não estiver em branco, você deverá tê-lo definido para CICS como uma transação local com um programa inicial que seja CSQCBP00. Esse campo se aplica apenas quando *UOWControl* possui o valor MQCUOWC\_FIRST ou MQCUOWC\_ONLY.

### ***FacilityLike (MQCHAR4) para MQCIH***

FacilityLike é o nome de um terminal instalado que deve ser usado como um modelo para o recurso de ponte

Um valor de espaços em branco significa que *FacilityLike* é obtido da definição de perfil de transação de ponte ou um valor padrão é usado...

Este campo é um campo de pedido utilizado apenas para transações 3270 O comprimento desse campo é fornecido por MQ\_FACILITY\_LIKE\_LENGTH O valor inicial deste campo é quatro espaços em branco.

### ***AttentionId (MQCHAR4) para MQCIH***

O valor neste campo determina o valor inicial da chave AID quando a transação é iniciada. É um valor de 1 byte, alinhado à esquerda.

AttentionId é um campo de solicitação usado apenas para transações 3270 O comprimento desse campo é fornecido por MQ\_ATTENTION\_ID\_LENGTH O valor inicial deste campo é quatro espaços em branco.

### ***StartCode (MQCHAR4) para MQCIH***

O valor desse campo é um indicador que especifica se a ponte emula uma transação de terminal ou uma transação iniciada com START.

O valor deve ser um dos seguintes:

#### **MQCSC\_START**

Iniciar.

#### **MQCSC\_STARTDATA**

Dados de início

#### **MQCSC\_TERMINPUT**

Entrada de terminal

#### **MQCSC\_NONE**

Nenhum.

Em todos os casos, para a linguagem de programação C, as constantes MQCSC\_\*\_ARRAY também são definidas; essas constantes têm os mesmos valores das constantes MQCSC\_\* correspondentes, mas são matrizes de caracteres em vez de sequências.

Na resposta da ponte, esse campo é configurado com o código de início apropriado para o próximo ID de transação contido no campo *NextTransactionId*. Os seguintes códigos de início são possíveis na resposta:

- MQCSC\_START
- MQCSC\_STARTDATA
- MQCSC\_TERMINPUT

Para CICS Transaction Server 1.2, esse campo é apenas um campo de solicitação; seu valor na resposta é indefinido.

Para o CICS Transaction Server 1.3 e liberações subsequentes, esse campo é um campo de pedido e de resposta.

Este campo é utilizado apenas para transações 3270. O comprimento desse campo é fornecido por MQ\_START\_CODE\_LENGTH. O valor inicial desse campo é MQCSC\_NONE.

### ***CancelCode (MQCHAR4) para MQCIH***

O valor neste campo é o código de encerramento anormal a ser utilizado para finalizar a transação (normalmente uma transação de conversação que está solicitando mais dados). Caso contrário, esse campo será configurado como espaços em branco.

Este campo é um campo de pedido utilizado apenas para transações 3270. O comprimento desse campo é fornecido por MQ\_CANCEL\_CODE\_LENGTH. O valor inicial deste campo é quatro espaços em branco.

### ***NextTransactionId (MQCHAR4) para MQCIH***

Esse valor é o nome da próxima transação retornada pela transação do usuário (geralmente por EXEC CICS RETURN TRANSID). Se não houver nenhuma próxima transação, esse campo será configurado como espaços em branco.

Este campo é um campo de resposta utilizado apenas para transações 3270. O comprimento desse campo é fornecido por MQ\_TRANSACTION\_ID\_LENGTH. O valor inicial deste campo é quatro espaços em branco.

### ***Reserved2 (MQCHAR8) para MQCIH***

Este campo é um campo reservado. O valor deve ser 8 espaços em branco.

### ***Reserved3 (MQCHAR8) para MQCIH***

Este campo é um campo reservado. O valor deve ser 8 espaços em branco.

### ***CursorPosition (MQLONG) para MQCIH***

O valor neste campo mostra a posição inicial do cursor quando a transação é iniciada.. Para transações conversacionais, a posição do cursor está no vetor RECEIVE.

Este campo é um campo de pedido utilizado apenas para transações 3270. O valor inicial desse campo é 0. Esse campo não estará presente se *Version* for menor que MQCIH\_VERSION\_2.

### ***ErrorOffset (MQLONG) para MQCIH***

O campo ErrorOffset mostra a posição de dados inválidos detectada pela saída da ponte. Este campo fornece o deslocamento do início da mensagem para o local dos dados inválidos.

ErrorOffset é um campo de resposta usado apenas para transações 3270. O valor inicial desse campo é 0. Esse campo não estará presente se *Version* for menor que MQCIH\_VERSION\_2.

### ***InputItem (MQLONG) para MQCIH***

Este campo é um campo reservado O valor deve ser 0.

Esse campo não estará presente se *Version* for menor que MQCIH\_VERSION\_2.

### ***Reserved4 (MQLONG) para MQCIH***

Este campo é um campo reservado O valor deve ser 0.

Esse campo não estará presente se *Version* for menor que MQCIH\_VERSION\_2.

## **MQCMHO-Criar opções de manipulação de mensagens**

A estrutura **MQCMHO** permite que os aplicativos especifiquem opções que controlam como as manipulações de mensagens são criadas A estrutura é um parâmetro de entrada na chamada **MQCRTMH ..**

### **Disponibilidade**

A estrutura **MQCMHO** está disponível nas plataformas a seguir:

-  AIX
-  IBM i
-  Linux
-  Windows
-  z/OS

e para o IBM MQ MQI clients conectado a esses sistemas

### **Conjunto de caracteres e codificação**

Os dados em **MQCMHO** devem estar no conjunto de caracteres do aplicativo e na codificação do aplicativo (**MQENC\_NATIVE**)..

### **Campos**

**Nota:** Na tabela a seguir, os campos são agrupados por uso e não alfabeticamente. Os tópicos filhos seguem a mesma sequência.

<i>Tabela 478. Campos em MQCMHO</i>		
<b>Nome e descrição do campo</b>	<b>Nome da constante</b>	<b>Valor inicial (se houver) da constante</b>
<u>StrucId</u> (identificador de estrutura)	MQCMHO_STRUC_ID	' CMHO '
<u>Versão</u> (número da versão da estrutura).	MQCMHO_VERSION_1	1
<u>Opções</u> (opções).	MQCMHO_DEFAULT_VAL IDATION	0
<b>Notas:</b>		
1. Na linguagem de programação C, a variável macroMQCMHO_DEFAULT contém os valores listados na tabela. Ele pode ser usado da seguinte maneira para fornecer valores iniciais para os campos na estrutura:		
<pre>MQCMHO MyCMHO = {MQCMHO_DEFAULT};</pre>		

## Declarações de idiomas

### Declaração C para MQCMHO

```
struct tagMQCMHO {
    MQCHAR4  StrucId;          /* Structure identifier */
    MQLONG   Version;         /* Structure version number */
    MQLONG   Options;         /* Options that control the action of MQCRTMH */
};
```

### Declaração COBOL para MQCMHO

```
** MQCMHO structure
10 MQCMHO.
** Structure identifier
15 MQCMHO-STRUCID PIC X(4).
** Structure version number
15 MQCMHO-VERSION PIC S9(9) BINARY.
** Options that control the action of MQCRTMH
15 MQCMHO-OPTIONS PIC S9(9) BINARY.
```

### Declaração PL/I para MQCMHO

```
dcl
1 MQCMHO based,
3 StrucId      char(4),      /* Structure identifier */
3 Version      fixed bin(31), /* Structure version number */
3 Options      fixed bin(31), /* Options that control the action of MQCRTMH */
```

### Declaração do High Level Assembler para MQCMHO

```
MQCMHO          DSECT
MQCMHO_STRUCID  DS CL4 Structure identifier
MQCMHO_VERSION DS F   Structure version number
MQCMHO_OPTIONS DS F   Options that control the action of
*               MQCRTMH
MQCMHO_LENGTH  EQU *-MQCMHO
MQCMHO_AREA    DS CL(MQCMHO_LENGTH)
```

### **StrucId (MQCHAR4) para MQCMHO.**

Este é o identificador de estrutura da estrutura de opções de manipulação de mensagens de criação.. Ele é sempre um campo de entrada Seu valor é MQCMHO\_STRUC\_ID.

O valor deve ser:.

#### **MQCMHO\_STRUC\_ID**

Identificador para criar estrutura de opções de manipulação de mensagem.

Para a linguagem de programação C, a constante **MQCMHO\_STRUC\_ID\_ARRAY** também é definida.. Isso tem o mesmo valor que **MQCMHO\_STRUC\_ID**, mas é uma matriz de caracteres em vez de uma sequência..

### **Versão (MQLONG) para MQCMHO**

Esse campo é sempre um campo de entrada Seu valor inicial é MQCMHO\_VERSION\_1.

Este é o número da versão da estrutura; o valor deve ser:

#### **MQCMHO\_VERSION\_1**

Version-1 criar estrutura de opções de manipulação de mensagens.

A constante a seguir especifica o número da versão atual:

## **MQCMHO\_CURRENT\_VERSION**

A versão atual da estrutura de opções de manipulação de mensagem de criação

## **Opções (MQLONG) para MQCMHO**

Esse campo é sempre um campo de entrada O valor inicial é MQCMHO\_DEFAULT\_VALIDATION.

Uma das seguintes opções pode ser especificada:

### **MQCMHO\_VALIDATE**

Quando **MQSETMP** é chamado para configurar uma propriedade nessa manipulação de mensagem, o nome da propriedade é validado para assegurar que:

- Não contém caracteres inválidos.
- não inicia JMS ou usr.JMS , exceto o seguinte:
  - JMSCorrelationID
  - JMSReplyTo
  - JMSType
  - JMSXGroupID
  - JMSXGroupSeq

Esses nomes são reservados para propriedades JMS .

- não é uma das seguintes palavras-chave, em qualquer mistura de maiúscula ou minúscula:
  - E
  - ENTRE
  - ESCAPE
  - FALSE
  - DENTRO
  - É
  - LIKE
  - NOT
  - NULL
  - OR
  - TRUE
- não começa Corpo. ou Raiz. (exceto para Root.MQMD.).

Se a propriedade for definida pelo MQ(mq. \*) e o nome for reconhecido, os campos do descritor de propriedades serão configurados para os valores corretos para a propriedade Se a propriedade não for reconhecida, o campo *Support* do descritor de propriedade será configurado como

### **MQPD\_OPTIONAL**

### **MQCMHO\_DEFAULT\_VALIDATION**

Este valor especifica que o nível padrão de validação de nomes de propriedades ocorre.

O nível padrão de validação é equivalente ao nível especificado por **MQCMHO\_VALIDATE**

Esse valor é o valor padrão.

### **MQCMHO\_NO\_VALIDATION**

Não ocorre nenhuma validação no nome da propriedade Consulte a descrição de **MQCMHO\_VALIDATE**

**Opção padrão:** Se nenhuma das opções anteriores descritas for necessária, a seguinte opção poderá ser usada:

## MQCMHO\_NONE

Todas as opções assumem seus valores padrão. Utilize este valor para indicar que nenhuma outra opção foi especificada.. **MQCMHO\_NONE** ajuda a documentação do programa; não se pretende que essa opção seja usada com nenhuma outra, mas como seu valor é zero, tal uso não pode ser detectado.

## MQCNO-Opções de conexão

A estrutura MQCNO permite que o aplicativo especifique opções relacionadas à conexão com o gerenciador de filas.. A estrutura é um parâmetro de entrada / saída na chamada MQCONNX.

Para obter mais informações sobre o uso de identificadores compartilhados e a chamada MQCONNX, consulte [Conexões compartilhadas \(independentes de encadeamento\) com MQCONNX](#)

## Disponibilidade

A estrutura MQCNO está disponível nas plataformas a seguir:

-  AIX
-  IBM i
-  Linux
-  Windows

e para o IBM MQ MQI clients conectado a esses sistemas

## Versão

Os arquivos de cabeçalho, COPY e INCLUDE fornecidos para as linguagens de programação suportadas contêm a versão mais recente de MQCNO, mas com o valor inicial do campo *Version* configurado como MQCNO\_VERSION\_1 Para usar os campos que não estão presentes na estrutura version-1 , o aplicativo deve configurar o campo *Version* para o número da versão necessário.

## Conjunto de caracteres e codificação

Os dados em MQCNO devem estar no conjunto de caracteres fornecido pelo atributo do gerenciador de fila do **CodedCharSetId** e a codificação do gerenciador de filas locais fornecido por MQENC\_NATIVE No entanto, se o aplicativo estiver em execução como um IBM MQ MQI client, a estrutura deverá estar no conjunto de caracteres e na codificação do cliente.

## Campos

**Nota:** Na tabela a seguir, os campos são agrupados por uso e não alfabeticamente. Os tópicos filhos seguem a mesma sequência.

Tabela 479. Campos em MQCNO para MQCNO		
Nome e descrição do campo	Nome da constante	Valor inicial (se houver) da constante
<u>StrucId</u> (identificador de estrutura)	MQCNO_STRUC_ID	'CNO~'
<u>Versão</u> (número da versão da estrutura).	MQCNO_VERSION_1	1
<u>Opções</u> (opções que controlam a ação de MQCONNX)	MQCNO_NONE	0
<b>Nota:</b> Os campos restantes serão ignorados se <i>Version</i> for menor que MQCNO_VERSION_2.		



Tabela 479. Campos em MQCNO para MQCNO (continuação)

Nome e descrição do campo	Nome da constante	Valor inicial (se houver) da constante
<u>ClientConnOffset</u> (deslocamento da estrutura MQCD para conexão do cliente)	Nenhum	0
<u>ClientConnPtr</u> (endereço da estrutura MQCD para conexão do cliente)	Nenhum	Ponteiro nulo ou bytes nulos
<b>Nota:</b> Os campos restantes são ignorados se <i>Version</i> for menor que MQCNO_VERSION_3.		
<u>ConnTag</u> (tag de conexão do gerenciador de filas)	MQCT_NONE	Nulos
<b>Nota:</b> Os campos restantes serão ignorados se o <i>Version</i> for menor que MQCNO_VERSION_4		
<u>SSLConfigPtr</u> (endereço da estrutura MQSCO para conexão do cliente)	Nenhum	Ponteiro nulo ou bytes nulos
<u>SSLConfigOffset</u> (deslocamento da estrutura MQSCO para conexão do cliente)	Nenhum	0
<b>Nota:</b> Os campos restantes serão ignorados se o <i>Version</i> for menor que MQCNO_VERSION_5		
<u>ConnectionId</u> (ID da conexão exclusivo).	Nenhum	Ponteiro nulo ou bytes nulos
<u>SecurityParmsOffset</u> (deslocamento da estrutura MQSCO para parâmetros de segurança)	Nenhum	Ponteiro nulo ou bytes nulos
<u>SecurityParmsPtr</u> (endereço da estrutura MQSCO para parâmetros de segurança)	Nenhum	Ponteiro nulo ou bytes nulos
<b>Nota:</b> Os campos restantes serão ignorados se <i>Version</i> for menor que MQCNO_VERSION_6.		
<u>Reservado</u> (campo reservado);	Nenhum	Campo reservado para preencher a estrutura para um limite de 64 bits.
<u>CCDTUrlLength</u> (CCDT URL length)	Nenhum	Comprimento da sequência identificada por <i>CCDTUrlPtr</i> ou <i>CCDTUrlOffset</i>
<u>CCDTUrlPtr</u> (ponteiro CCDT URL )	Nenhum	Ponteiro para uma sequência que contém uma URL, para identificar o local da tabela do canal de conexão do cliente para usar para a conexão.
<u>CCDTUrlOffset</u> (deslocamento de CCDT URL )	Nenhum	Deslocamento em bytes de uma sequência que contém uma URL que identifica o local da tabela de canais de conexão do cliente a ser usado para a conexão..

Tabela 479. Campos em MQCNO para MQCNO (continuação)

Nome e descrição do campo	Nome da constante	Valor inicial (se houver) da constante
<b>Nota:</b> Os campos restantes serão ignorados se <i>Version</i> for menor do que MQCNO_VERSION_7		
<u>ApplName</u> (nome configurado pelo aplicativo)	Nenhum	Nome configurado pelo aplicativo para identificar a conexão com o gerenciador de filas
<u>Reserved2</u> (campo reservado)	Nenhum	Campo reservado para preencher a estrutura para um limite de 64 bits.
<div style="background-color: #e0e0e0; padding: 2px; margin-bottom: 5px;"> <span style="background-color: #000080; color: white; padding: 2px;">&gt; V9.3.0</span> <span style="background-color: #000080; color: white; padding: 2px;">&gt; V9.3.0</span> </div> <b>Nota:</b> Os campos restantes serão ignorados se <i>Version</i> for menor que MQCNO_VERSION_8.		
<div style="background-color: #e0e0e0; padding: 2px; display: inline-block;">&gt; V9.3.0</div> <u>BalanceParms</u>   <u>Offset</u>	Nenhum	Deslocamento em bytes para a estrutura MQBNO
<div style="background-color: #e0e0e0; padding: 2px; display: inline-block;">&gt; V9.3.0</div> <u>BalanceParmsPtr</u>	Nenhum	Ponteiro para o local da estrutura MQBNO
<b>Notas:</b> <ol style="list-style-type: none"> <li>1. O símbolo ~ representa um único caractere em branco.</li> <li>2. Na linguagem de programação C, a variável macroMQCNO_DEFAULT contém os valores que estão listados na tabela. Use-o da seguinte maneira para fornecer valores iniciais para os campos na estrutura:             <div style="background-color: #e0e0e0; padding: 5px; margin-top: 10px;"> <pre>MQCNO MyCNO = {MQCNO_DEFAULT};</pre> </div> </li> </ol>		

## Declarações de idiomas

**Nota:**

> V9.3.0

 Em cada uma das declarações a seguir, as duas últimas linhas (*Offset of the MQBMO structure* e *Address of the location of the MQBMO structure*) foram incluídas em IBM MQ 9.2.4 para usuários CD e em IBM MQ 9.3.0 para usuários LTS.

Declaração C para MQCNO

```
typedef struct tagMQCNO MQCNO;
struct tagMQCNO {
    MQCHAR4    StructId;           /* Structure identifier */
    MQLONG     Version;           /* Structure version number */
    MQLONG     Options;           /* Options that control the action of
    MQCONNXX */
    MQLONG     ClientConnOffset; /* Offset of MQCD structure for client
    connection */
    MQPTR      ClientConnPtr;     /* Address of MQCD structure for client
    connection */
    MQBYTE128  ConnTag;           /* Queue manager connection tag */
    PMQSCO     SSLConfigPtr;      /* Address of MQSCO structure for client
    connection */
    MQLONG     SSLConfigOffset; /* Offset of MQSCO structure for client
    connection */
    MQBYTE24   ConnectionId;      /* Unique connection identifier */
    MQLONG     SecurityParmsOffset /* Security fields */
    PMQCSP     SecurityParmsPtr /* Security parameters */
};
```

```

MQLONG    CCDTurlLength      /* Length of string identified by Ptr or offset */
MQLONG    CCDTurlOffset     /* Offset in bytes to URL of client connection channel */
PMQURL    CCDTurlPtr        /* Address of string containing URL */
MQBYTE4   Reserved         /* Reserved field to pad out to 64 bit boundary */
MQCHAR28  ApplName         /* Name set by the application to identify the connection to
                           the queue manager */
MQBYTE4   Reserved2        /* Reserved field to pad out to 64 bit boundary */
MQLONG    BalanceParmsOffset /* Offset of the MQBMO structure */
PMQBMO    BalanceParmsPtr   /* Address of the location of the MQBMO structure */
};

```

## Declaração COBOL para MQCNO

```

** MQCNO structure
10 MQCNO.
** Structure identifier
15 MQCNO-STRUCID PIC X(4).
** Structure version number
15 MQCNO-VERSION PIC S9(9) BINARY.
** Options that control the action of MQCONN
15 MQCNO-OPTIONS PIC S9(9) BINARY.
** Offset of MQCD structure for client connection
15 MQCNO-CLIENTCONNOFFSET PIC S9(9) BINARY.
** Address of MQCD structure for client connection
15 MQCNO-CLIENTCONNPTR POINTER.
** Queue manager connection tag
15 MQCNO-CONN TAG PIC X(128).
** Address of MQSCO structure for client connection
15 MQCNO-SSLCONFIGPTR POINTER.
** Offset of MQSCO structure for client connection
15 MQCNO-SSLCONFIGOFFSET PIC S9(9) BINARY.
** Unique connection identifier
15 MQCNO-CONNECTIONID PIC X(24).
** Offset of MQCSP structure for security parameters
15 MQCNO-SECURITYPARMSOFFSET PIC S9(9) BINARY.
** Address of MQCSP structure for security parameters
15 MQCNO-SECURITYPARMSPTR POINTER.
** Length of string identified by CCDTURLOFFSET or CCDTURLPTR
15 MQCNO-CCDTURLLENGTH
** Pointer to a string which contains a URL, to identify the location of the client
connection channel
15 MQCNO-CCDTURLPTR
** Address of string which contains a URL that identifies the location of the client
connection channel table
15 MQCNO-CCDTURLOFFSET
** Reserved field to pad to 64 bit boundary
15 MQCNO-RESERVED
** Name set by the application to identify the connection to the queue manager
15 MQCNO-APPLNAME
** Reserved field to pad to 64 bit boundary
15 MQCNO-RESERVED2
** Address of the MQBMO structure
15 MQCNO-BALANCEPARMSOFFSET
** Pointer to the MQBMO structure
15 MQCNO-BALANCEPARMSPTR

```

## Declaração PL/I para MQCNO

```

dcl
1 MQCNO based,
3 StrucId          char(4),          /* Structure identifier */
3 Version          fixed bin(31),   /* Structure version number */
3 Options          fixed bin(31),   /* Options that control the action
of MQCONN */
3 ClientConnOffset fixed bin(31),   /* Offset of MQCD structure for
client connection */
3 ClientConnPtr    pointer,         /* Address of MQCD structure for
client connection */
3 ConnTag          char(128),       /* Queue manager connection tag */
3 SSLConfigPtr     pointer,         /* Address of MQSCO structure for
client connection */
3 SSLConfigOffset  fixed bin(31),   /* Offset of MQSCO structure for
client connection */
3 ConnectionId     char(24),        /* Unique connection identifier
3 SecurityParmsOffset fixed bin(31) /* Offset of MQCSP structure for
security parameters */
3 SecurityParmsPtr pointer,         /* Address of MQCSP structure for

```

3	CCDTUrlLength	fixed bin(31)	security parameters */ /* Length of string identified by <i>CCDTUrlPtr</i> or <i>CCDTUrlOffset</i> */
3	CCDTUrlOffset	fixed bin(31)	/* Offset in bytes to URL of client connection channel */
3	CCDTUrlPtr	pointer	/* Pointer to string containing URL */
3	Reserved	char(4)	/* Reserved field to pad out to 64 bit boundary */
3	AppName	char(28)	/* Name set by the application to identify the connection to
3	Reserved2	char(4)	the queue manager */ /* Reserved field to pad out to 64 bit boundary */
3	BalanceParmsOffset	fixed bin(31)	/* Offset of the MQBMO structure */
3	BalanceParmsPtr	pointer	/* Address of the MQBMO structure */

## Declaração High Level Assembler para MQCNO

MQCNO	DSECT		
MQCNO_STRUCID	DS	CL4	Structure identifier
MQCNO_VERSION	DS	F	Structure version number
MQCNO_OPTIONS	DS	F	Options that control the action of MQCONN
* MQCNO_CLIENTCONNOFFSET	DS	F	Offset of MQCD structure for client connection
* MQCNO_CLIENTCONNPTR	DS	F	Address of MQCD structure for client connection
* MQCNO_CONNTAG	DS	XL128	Queue manager connection tag
MQCNO_CONNECTIONID	DS	XL24	Unique connection identifier
MQCNO_SSLCONFIGOFFSET	DS	F	Offset of MQCSP structure for security parameters
* MQCNO_SSLCONFIGPTR	DS	F	Address of MQCSP structure for security parameters
* MQCNO_LENGTH	EQU	*-MQCNO	
	ORG	MQCNO	
MQCNO_AREA	DS	CL(MQCNO_LENGTH)	
MQCNO_CCDTURLLENGTH	DS	F	Length of string identified by <i>CCDTURLPTR</i> or <i>CCDTURLOFFSET</i>
* MQCNO_CCDTURLOFFSET	DS	F	Offset in bytes to URL of client connection channel
MQCNO_CCDTURLPTR	DS	F	Pointer to string containing URL
RESERVED	DS	XL4	Reserved field to pad out to 64 bit boundary
APPLNAME	DS	CL28	Name set by the application to identify the connection to the queue manager
* RESERVED2	DS	XL4	Reserved field to pad out to 64 bit boundary
MQCNO_BALANCEPARMSOFFSET	DS	F	Offset of the MQBMO structure
MQCNO_BALANCEPARMSPTR	DS	F	Address of the MQBMO structure

## Declaração Visual Basic para MQCNO

Type MQCNO		
StrucId	As String*4	'Structure identifier'
Version	As Long	'Structure version number'
Options	As Long	'Options that control the action of' 'MQCONN'
ClientConnOffset	As Long	'Offset of MQCD structure for client' 'connection'
ClientConnPtr	As MQPTR	'Address of MQCD structure for client' 'connection'
ConnTag	As MQBYTE128	'Queue manager connection tag'
SSLConfigPtr	As MQPTR	'Address of MQSCO structure for client' 'connection'
SSLConfigOffset	As Long	'Offset of MQSCO structure for client' 'connection'
ConnectionId	As MQBYTE24	'Unique connection identifier'
SecurityParmsOffset	As Long	'Offset of MQCSP structure for security' 'parameters'
SecurityParmsPtr	As MQPTR	'Address of MQCSP structure for security' 'parameters'
CCDTUrlLength	As Long	'Length of string identified by <i>CCDTUrlPtr</i> ' 'or <i>CCDTUrlOffset</i> '
CCDTUrlOffset	As Long	'Offset in bytes to URL of client connection channel'
CCDTUrlPtr	As MQPTR	'Pointer to string containing URL'
Reserved	As MQBYTE4	'Reserved field to pad out to 64 bit boundary'
AppName	As String*28	'Name set by the application to identify the connection to' 'the queue manager'
Reserved2	As MQBYTE4	'Reserved field to pad out to 64 bit boundary'
BalanceParmsOffset	As Long	'Offset in bytes to MQBMO structure'

BalanceParamsPtr	As MQPTR	'Address of MQBNO structure'
End Type		

## Tarefas relacionadas

[Usando MQCONN](#)

### **StrucId (MQCHAR4) para MQCNO**

Este é o identificador da estrutura de opções de conexão. Ele é sempre um campo de entrada Seu valor é MQCNO\_STRUC\_ID.

O valor deve ser:.

#### **MQCNO\_STRUC\_ID**

Identificador para a estrutura de opções de conexão.

Para a linguagem de programação C, a constante MQCNO\_STRUC\_ID\_ARRAY também é definida. Essa constante tem o mesmo valor que MQCNO\_STRUC\_ID, mas é uma matriz de caracteres em vez de uma cadeia.

### **Versão (MQLONG) para MQCNO**

A versão é sempre um campo de entrada Seu valor inicial é MQCNO\_VERSION\_1.

O valor deve ser um dos seguintes:

#### **MQCNO\_VERSION\_1**

Estrutura de opções de conexão Version-1 .

#### **MQCNO\_VERSION\_2**

Estrutura de opções de conexão Version-2 .

#### **MQCNO\_VERSION\_3**

Estrutura de opções de conexão da Version-3

#### **MQCNO\_VERSION\_4**

Estrutura de opções de conexão Version-4 .

#### **MQCNO\_VERSION\_5**

Estrutura de opções de conexão Version-5 .

#### **MQCNO\_VERSION\_6**

Estrutura de opções de conexão Version-6 .

#### **MQCNO\_VERSION\_7**

Estrutura de opções de conexão Version-7 .

#### **V9.3.0 MQCNO\_VERSION\_8**

Estrutura de opções de conexão Version-8 .

Os campos que existem apenas nas versões mais recentes da estrutura são identificados como tal nas descrições dos campos A constante a seguir especifica o número da versão atual:

#### **MQCNO\_CURRENT\_VERSION**

A versão atual da estrutura de opções de conexão.

### **Opções (MQLONG) para MQCNO**

Opções que controlam a ação de MQCONN.

## **Opções de contabilidade**

As opções a seguir controlam o tipo de contabilidade se o atributo do gerenciador de filas **AccountingConnOverride** estiver configurado como MQMON\_ENABLED:

### **MQCNO\_ACCOUNTING\_MQI\_ENABLED**

Quando a coleta de dados de monitoramento é desativada na definição do gerenciador de filas configurando o atributo **MQIAccounting** como MQMON\_OFF, a configuração desse sinalizador ativa a coleta de dados contábeis do MQI

### **MQCNO\_ACCOUNTING\_MQI\_DISABLED**

Quando a coleta de dados de monitoramento é desativada na definição do gerenciador de filas configurando o atributo **MQIAccounting** como MQMON\_OFF, configurar esse sinalizador para a coleta de dados contábeis do MQI

### **MQCNO\_ACCOUNTING\_Q\_ENABLED**

Quando a coleta de dados de contabilidade de filas é desativada na definição de gerenciador de filas configurando o atributo **MQIAccounting** como MQMON\_OFF, configurar essa sinalização ativa a coleta de dados de contabilidade para aquelas filas que especificam um gerenciador de filas no campo *MQIAccounting* de sua definição de fila.

### **MQCNO\_ACCOUNTING\_Q\_DISABLED**

Quando a coleta de dados de contabilidade de filas é desativada na definição de gerenciador de filas configurando o atributo **MQIAccounting** como MQMON\_OFF, a configuração desse sinalizador desativa a coleta de dados de contabilidade para as filas que especificam um gerenciador de fila no campo *MQIAccounting* de sua definição de fila

Se nenhum desses sinalizadores for definido, a contabilidade para a conexão será conforme definido nos atributos do gerenciador de fila..

## **Opções de Ligação**

As seguintes opções controlam o tipo de ligação IBM MQ a ser usado. Especifique apenas uma destas opções:

### **MQCNO\_STANDARD\_BINDING**

O aplicativo e o agente do gerenciador de filas locais (o componente que gerencia operações de enfileiramento) são executados em unidades separadas de execução (geralmente, em processos separados). Este acordo mantém a integridade do gerenciador de filas, ou seja, ele protege o gerenciador de fila de programas errantes.

Se o gerenciador de filas suportar diversos tipos de ligação e você configurar MQCNO\_STANDARD\_BINDING, o gerenciador de fila usará o atributo **DefaultBindType** na sub-rotina *Connection* no arquivo *qm.ini* para selecionar o tipo real de ligação.. Se essa sub-rotina não for definida ou o valor não puder ser usado ou não for apropriado para o aplicativo, o gerenciador de filas selecionará um tipo de ligação apropriado. O gerenciador de filas configura o tipo de ligação real utilizado nas opções de conexão

Use MQCNO\_STANDARD\_BINDING em situações em que o aplicativo pode não ter sido totalmente testado, ou pode não ser confiável ou não confiável MQCNO\_STANDARD\_BINDING é o padrão.

Essa opção é suportada em todos os ambientes

Se estiver vinculando à biblioteca *mqm*, uma conexão do servidor padrão usando o tipo de ligação padrão será tentada primeiro. Se o carregamento da biblioteca do servidor subjacente tiver falhado, uma conexão do cliente será tentada ao invés.

- Para alterar o comportamento de MQCONN (ou MQCONNX se MQCNO\_STANDARD\_BINDING for especificado), configure a variável de ambiente MQ\_CONNECT\_TYPE para uma das opções a seguir: Observe que há uma exceção para isso: se MQCNO\_FASTPATH\_BINDING for especificado com MQ\_CONNECT\_TYPE configurado como LOCAL ou STANDARD, as conexões de atalho poderão ser rebaixados pelo administrador sem uma mudança relacionada ao aplicativo.

Value	Significado
CLIENTE	É tentada apenas uma conexão do cliente.

Tabela 480. Valores para MQ\_CONNECT\_TYPE que mudam o comportamento de MQCONN ou MQCONNX (continuação)

Value	Significado
FASTPATH	Esse valor era suportado em liberações anteriores, mas agora ele será ignorado se for especificado.
LOCAL	É tentada apenas uma conexão do servidor. Conexões de atalho são transferidas por downgrade para uma conexão padrão do servidor.
STANDARD	Suportado para compatibilidade com liberações anteriores. Esse valor agora é tratado como LOCAL.

- Se a variável de ambiente MQ\_CONNECT\_TYPE não for configurada quando MQCONNX for chamado, uma conexão do servidor padrão usando o tipo de ligação padrão será tentada.. Se o carregamento da biblioteca do servidor tiver falhado, uma conexão do cliente será tentada.




### MQCNO\_FASTPATH\_BINDING

O aplicativo e o agente do gerenciador de filas locais fazem parte da mesma unidade de execução.. Isso está em contraste com o método típico de ligação, em que o aplicativo e o agente do gerenciador de fila local são executados em unidades separadas de execução


MQCNO\_FASTPATH\_BINDING será ignorado se o gerenciador de filas não suportar este tipo de ligação; o processamento continua como se a opção não tivesse sido especificada

MQCNO\_FASTPATH\_BINDING pode ser vantajoso em situações em que vários processos consomem mais recursos do que o recurso geral usado pelo aplicativo Um aplicativo que usa a ligação de atalho é conhecido como um *aplicativo confiável*

Considere os seguintes pontos importantes ao decidir se deve usar a ligação de atalho:

- Usar a opção MQCNO\_FASTPATH\_BINDING não impede que um aplicativo altere ou corrompa mensagens e outras áreas de dados pertencentes ao gerenciador de filas. Use esta opção apenas em situações em que você avaliou totalmente esses problemas
- O aplicativo não deve usar sinais assíncronos ou interrupções do cronômetro (como sigkill) com MQCNO\_FASTPATH\_BINDING. Há também restrições no uso de segmentos de memória compartilhada.
- O aplicativo deve usar a chamada MQDISC para desconectar do gerenciador de filas.
- O aplicativo deve ser concluído antes do término do gerenciador de fila com o comando endmqm
-  No IBM i, a tarefa deve ser executada sob um perfil do usuário que pertença ao grupo QMQMADM Além disso, o programa não deve parar de forma anormal, caso contrário resultados imprevisíveis podem ocorrer.
-   No AIX and Linux, o identificador de usuário mqm deve ser o identificador de usuário efetivo e o identificador de grupo mqm deve ser o identificador de grupo efetivo Para fazer o aplicativo ser executado dessa maneira, configure o programa para que ele seja de propriedade do mqm identificador de usuário e mqm identificador de grupo e, em seguida, configure os bits de permissão setuid e setgid no programa.

O Object Authority Manager (OAM) do IBM MQ ainda usa o ID do usuário real para verificação de autoridade

-  No Windows, o programa deve ser um membro do grupo mqm A ligação de atalho não é suportada para aplicativos de 64 bits.

A opção MQCNO\_FASTPATH\_BINDING é suportada nos ambientes a seguir:

-  AIX

-  IBM i
-  Linux
-  Windows

 No z/OS, a opção é aceita mas ignorada.

Para obter mais informações sobre as implicações de usar aplicativos confiáveis, consulte [Restrições para aplicativos confiáveis](#).

### **MQCNO\_SHARED\_BINDING**


MQCNO\_SHARED\_BINDING, o aplicativo e o agente do gerenciador de filas locais compartilham alguns recursos. MQCNO\_SHARED\_BINDING será ignorado se o gerenciador de filas não suportar esse tipo de ligação. O processamento continuará, embora a opção não tenha sido especificada.

### **MQCNO\_ISOLATED\_BINDING**

Nesse caso, o processo de aplicativo e o agente do gerenciador da fila local são isolados uns dos outros, pois não compartilham recursos. MQCNO\_ISOLATED\_BINDING será ignorada se o gerenciador de filas não suportar esse tipo de ligação. O processamento continuará, embora a opção não tenha sido especificada.


### **MQCNO\_CLIENT\_BINDING**

Especifique essa opção para que o aplicativo tente apenas uma conexão do cliente. Essa opção tem as seguintes limitações:

-  MQCNO\_CLIENT\_BINDING é ignorado no z/OS.
- MQCNO\_CLIENT\_BINDING é rejeitado com MQRC\_OPTIONS\_ERROR se for especificado com qualquer opção de ligação MQCNO que não MQCNO\_STANDARD\_BINDING.
- MQCNO\_CLIENT\_BINDING não está disponível para Java ou .NET já que eles têm seus próprios mecanismos para escolher o tipo de ligação.

### **MQCNO\_LOCAL\_BINDING**

Especifique essa opção para que o aplicativo tente uma conexão do servidor. Se MQCNO\_FASTPATH\_BINDING, MQCNO\_ISOLATED\_BINDING ou MQCNO\_SHARED\_BINDING também estiverem especificadas, então ao invés a conexão será desse tipo e será documentada nesta seção. Caso contrário, uma conexão do servidor padrão será tentada usando o tipo de ligação padrão. MQCNO\_LOCAL\_BINDING tem as seguintes limitações:

-  MQCNO\_LOCAL\_BINDING é ignorado em z/OS.
- MQCNO\_LOCAL\_BINDING é rejeitado com MQRC\_OPTIONS\_ERROR se for especificado com qualquer opção de reconexão MQCNO além de MQCNO\_RECONNECT\_AS\_DEF.
- MQCNO\_LOCAL\_BINDING não está disponível para Java ou .NET já que eles têm seus próprios mecanismos para escolher o tipo de ligação.

Nas plataformas a seguir, é possível usar a variável de ambiente MQ\_CONNECT\_TYPE com o tipo de ligação especificado pelo campo Options para controlar o tipo de ligação usado.

-  AIX
-  Linux
-  Windows

Se você especificar essa variável de ambiente, ela deverá ter o valor FASTPATH ou STANDARD ; se tiver um valor diferente, ele será ignorado. O valor da variável de ambiente faz distinção entre maiúsculas e minúsculas; consulte [MQCONN environment variable](#) para obter mais informações.



A variável de ambiente e o campo *Options* interagem como a seguir:

- Se você omitir a variável de ambiente ou fornecer a ela um valor que não seja suportado, o uso da ligação de atalho será determinado exclusivamente pelo campo *Options*.
- Se você fornecer à variável de ambiente um valor suportado, a ligação de atalho será usada apenas se a variável de ambiente e o campo *Options* especificarem a ligação de atalho...

## Opções de tag de conexão.

Essas opções são suportadas apenas ao se conectar a um gerenciador de filas do z/OS e controlam o uso da tag de conexão *ConnTag*. É possível especificar apenas uma destas opções

A implementação precisa de tags de conexão difere entre IBM MQ for z/OS e IBM MQ for Multiplatforms:

- **z/OS** As opções a seguir, além de *MQCNO\_GENERATE\_CONN\_TAG*, são suportadas apenas ao conectar-se a um gerenciador de filas do z/OS e elas controlam o uso da tag de conexão. É possível especificar apenas uma das opções suportadas
- **ALW** *MQCNO\_GENERATE\_CONN\_TA* é suportado apenas em plataformas diferentes de z/OS.

### **ALW** *MQCNO\_GENERATE\_CONN\_TAG*

Retorna a tag de conexão que o gerenciador de fila associou a essa conexão, na estrutura *MQCNO* de entrada.

A tag de conexão retornada será idêntica para todas as conexões que o gerenciador de filas considera como uma única Instância do aplicativo

### **z/OS** *MQCNO\_SERIALIZE\_CONN\_TAG\_Q\_MGR*

Esta opção solicita uso exclusivo da tag de conexão no gerenciador de filas locais. Se a tag de conexão já estiver em uso no gerenciador de fila local, a chamada *MQCONN* falhará com o código de razão *MQRC\_CONN\_TAG\_IN\_USE*. O resultado da chamada não é afetado pelo uso da tag de conexão em outro lugar no grupo de filas compartilhadas ao qual o gerenciador de filas locais pertence

### **z/OS** *MQCNO\_SERIALIZE\_CONN\_TAG\_QSG*

Essa opção solicita o uso exclusivo da tag de conexão dentro do grupo de filas compartilhadas ao qual o gerenciador de filas locais pertence. Se a tag de conexão já estiver em uso no grupo de filas compartilhadas, a chamada *MQCONN* falhará com o código de razão *MQRC\_CONN\_TAG\_IN\_USE*

### **z/OS** *MQCNO\_RESTRICT\_CONN\_TAG\_Q\_MGR*

Essa opção solicita o uso compartilhado da tag de conexão dentro do gerenciador de fila local. Se a tag de conexão já estiver em uso no gerenciador de filas locais, a chamada *MQCONN* poderá ser bem-sucedida se o aplicativo solicitante estiver em execução no mesmo escopo do processamento do usuário existente da tag. Se essa condição não for satisfeita, a chamada *MQCONN* falhará com o código de razão *MQRC\_CONN\_TAG\_IN\_USE*. O resultado da chamada não é afetado pelo uso da tag de conexão em outro lugar no grupo de filas compartilhadas ao qual o gerenciador de fila local pertence

- Os aplicativos devem ser executados no mesmo espaço de endereço MVS para compartilhar a tag de conexão. Se o aplicativo usando a tag de conexão for um aplicativo cliente, *MQCNO\_RESTRICT\_CONN\_TAG\_Q\_MGR* não será permitido.

### **z/OS** *MQCNO\_RESTRICT\_CONN\_TAG\_QSG*

Essa opção solicita o uso compartilhado da tag de conexão dentro do grupo de filas compartilhadas ao qual o gerenciador de filas locais pertence. Se a tag de conexão já estiver em uso no grupo de filas compartilhadas, a chamada *MQCONN* poderá ser bem-sucedida desde que o aplicativo solicitante esteja em execução no mesmo escopo de processamento e conectado ao mesmo gerenciador de filas, como o usuário existente da tag.

Se essas condições não forem satisfeitas, a chamada MQCONNX falhará com o código de razão MQRC\_CONN\_TAG\_IN\_USE.

- Os aplicativos devem ser executados no mesmo espaço de endereço MVS para compartilhar a tag de conexão. Se o aplicativo usando a tag de conexão for um aplicativo cliente, MQCNO\_RESTRICT\_CONN\_TAG\_QSG não será permitido.

Se nenhuma dessas opções for especificada, ConnTag não será usado.. Essas opções não serão válidas se o `Version` for menor que MQCNO\_VERSION\_3

## Opções de compartilhamento de manipulação

### Multi

Essas opções são suportadas nos seguintes ambientes:

-  AIX
-  IBM i
-  Linux
-  Windows

Eles controlam o compartilhamento de identificadores entre diferentes encadeamentos (unidades de processamento paralelo) no mesmo processo. É possível especificar apenas uma destas opções:

### MQCNO\_HANDLE\_SHARE\_NONE

Essa opção indica que a conexão e os identificadores de objetos podem ser usados apenas pelo encadeamento que causou a alocação do identificador (ou seja, o encadeamento que emitiu a chamada MQCONN, MQCONNX ou MQOPEN). Os identificadores não podem ser usados por outros encadeamentos pertencentes ao mesmo processo

### MQCNO\_HANDLE\_SHARE\_BLOCK

Essa opção indica que os identificadores de conexão e de objeto alocados por um encadeamento de um processo podem ser usados por outros encadeamentos pertencentes ao mesmo processo. No entanto, apenas um encadeamento por vez pode usar qualquer identificador específico; ou seja, apenas o uso serial de um manipulador é permitido. Se um encadeamento tentar usar um identificador que já esteja em uso por outro encadeamento, os blocos de chamada (espera) até que o identificador se torne disponível.

### MQCNO\_HANDLE\_SHARE\_NO\_BLOCK

Isso é o mesmo que MQCNO\_HANDLE\_SHARE\_BLOCK, exceto que se o identificador estiver em uso por outro encadeamento, a chamada será concluída imediatamente com MQCC\_FAILED e MQRC\_CALL\_IN\_PROGRESS em vez de bloquear até que o identificador se torne disponível.

Um encadeamento pode ter zero ou um identificador não compartilhado:

- Cada chamada MQCONN ou MQCONNX que especifica MQCNO\_HANDLE\_SHARE\_NONE retorna um novo identificador não compartilhado na primeira chamada e o mesmo identificador não compartilhado na segunda e posterior chamadas (assumindo nenhuma chamada MQDISC de intervenção). O código de razão é MQRC\_ALREADY\_CONNECTED para a segunda e posterior chamadas.
- Cada chamada MQCONNX que especifica MQCNO\_HANDLE\_SHARE\_BLOCK ou MQCNO\_HANDLE\_SHARE\_NO\_BLOCK retorna uma nova manipulação compartilhada em cada chamada.

Os identificadores de objeto herdam as mesmas propriedades de compartilhamento que o identificador de conexão especificado na chamada MQOPEN que criou o identificador de objeto. Além disso, as unidades de trabalho herdam as mesmas propriedades de compartilhamento que o identificador de

conexão usado para iniciar a unidade de trabalho; se a unidade de trabalho for iniciada em um encadeamento usando um identificador compartilhado, a unidade de trabalho poderá ser atualizada em outro encadeamento usando o mesmo identificador

Se você não especificar uma opção de compartilhamento de manipulação, o padrão será determinado pelo ambiente:

- **Windows** No ambiente Microsoft Transaction Server (MTS), o padrão é igual a MQCNO\_HANDLE\_SHARE\_BLOCK.
- Em outros ambientes, o padrão é o mesmo que MQCNO\_HANDLE\_SHARE\_NONE

## Opções de reconexão.

Opções de reconexão determinam se uma conexão é reconectável. Apenas conexões do cliente são reconectáveis.

### MQCNO\_RECONNECT\_AS\_DEF

A opção de reconexão é resolvida para seu valor padrão. Se nenhum padrão for configurado, o valor dessa opção será resolvido para DISABLED. O valor da opção é transmitido ao servidor e pode ser consultado por PCF e MQSC.

### MQCNO\_RECONNECT

O aplicativo pode ser reconectado a qualquer gerenciador de fila consistente com o valor do parâmetro **QmgrName** de MQCONNX. Use a opção MQCNO\_RECONNECT somente se não houver nenhuma afinidade entre o aplicativo cliente e o gerenciador de fila com o qual ele estabeleceu inicialmente uma conexão. O valor da opção é transmitido ao servidor e pode ser consultado por PCF e MQSC.

### MQCNO\_RECONNECT\_DISABLED

O aplicativo não pode ser reconectado.. O valor da opção não é transmitido ao servidor.

### MQCNO\_RECONNECT\_Q\_MGR

O aplicativo pode ser reconectado somente ao gerenciador de filas com o qual ele foi conectado originalmente. Use este valor se um cliente puder ser reconectado, mas houver uma afinidade entre o aplicativo cliente e o gerenciador de filas com o qual ele estabeleceu originalmente uma conexão. Escolha este valor se você deseja que um cliente restabeleça automaticamente a conexão com a instância em espera de um gerenciador de filas altamente disponível. O valor da opção é transmitido ao servidor e pode ser consultado por PCF e MQSC.

Use as opções MQCNO\_RECONNECT, MQCNO\_RECONNECT\_DISABLED e MQCNO\_RECONNECT\_Q\_MGR apenas para conexões do cliente.. Se as opções forem usadas para uma conexão de ligação, o MQCONNX falhará com o código de conclusão MQCC\_FAILED e o código de razão MQRC\_OPTIONS\_ERROR. A reconexão automática do cliente não é suportada pelo IBM MQ classes for Java ..

## Opções de compartilhamento de conversa

As opções a seguir aplicam-se apenas a conexões do cliente TCP/IP. Para canais SNA, SPX e NetBios, esses valores são ignorados e o canal é executado como em versões anteriores do produto.

### MQCNO\_NO\_CONV\_SHARING

Esta opção não permite o compartilhamento de conversação.

Você pode usar MQCNO\_NO\_CONV\_SHARING em situações em que as conversas são muito carregadas e, portanto, em que a contenção é uma possibilidade na extremidade da conexão do servidor da instância do canal na qual as conversas de compartilhamento existem.

MQCNO\_NO\_CONV\_SHARING se comporta como SHARECNV (1) quando conectado a um canal que

suporta compartilhamento de conversa e SHARECNV (0) quando conectado a um canal que não suporta compartilhamento de conversa.

### **MQCNO\_ALL\_CONVS\_SHARE**

Essa opção permite o compartilhamento de conversas; o aplicativo não coloca nenhum limite no número de conexões na instância do canal. Esta opção é o valor padrão.

Se o aplicativo indicar que a instância do canal pode compartilhar, mas a definição de *SharingConversations* (SHARECNV) na extremidade da conexão do servidor do canal for configurada como uma, nenhum compartilhamento ocorrerá e nenhum aviso será fornecido ao aplicativo.

Da mesma forma, se o aplicativo indicar que o compartilhamento é permitido, mas a definição *SharingConversations* de conexão do servidor for configurada como zero, nenhum aviso será fornecido e o aplicativo exibirá o mesmo comportamento que um cliente em versões anteriores a IBM WebSphere MQ 7.0; a configuração do aplicativo relacionada a conversas de compartilhamento será ignorada

MQCNO\_NO\_CONV\_SHARING e MQCNO\_ALL\_CONVS\_SHARE são mutuamente exclusivos. Se ambas as opções forem especificadas em uma conexão específica, a conexão será rejeitada com um código de razão de MQRC\_OPTIONS\_ERROR

## **Opções de definição de canal**

As opções a seguir controlam o uso da estrutura de definição de canal transmitida no MQCNO:

### **MQCNO\_CD\_FOR\_OUTPUT\_ONLY**

Essa opção permite que a estrutura de definição de canal no MQCNO seja usada apenas para retornar o nome do canal usado em uma chamada MQCONN bem-sucedida.

Se uma estrutura de definição de canal válida não for fornecida, a chamada falhará com o código de razão MQRC\_CD\_ERROR.

Se o aplicativo não estiver em execução como um cliente, a opção será ignorada

O nome do canal retornado pode ser usado em uma chamada MQCONN subsequente usando a opção MQCNO\_USE\_CD\_SELECTION para reconectar usando a mesma definição de canal. Isso pode ser útil quando há várias definições de canal aplicáveis na tabela de canal do cliente...

### **MQCNO\_USE\_CD\_SELECTION**

Essa opção permite que a chamada MQCONN se conecte usando o nome do canal contido na estrutura de definição de canal transmitida no MQCNO..

Se a variável de ambiente MQSERVER for configurada a definição de canal definida por ela será usada. Se MQSERVER não for configurado, a tabela de canal do cliente será usada...

Se uma definição de canal com o nome do canal correspondente e o nome do gerenciador de filas não for localizada, a chamada falhará com código de razão MQRC\_Q\_MGR\_NAME\_ERROR.

Se uma estrutura de definição de canal válida não for fornecida, a chamada falhará com o código de razão MQRC\_CD\_ERROR.

Se o aplicativo não estiver em execução como um cliente, a opção será ignorada

## **Opção Padrão**

Se você não precisar de nenhuma das opções descritas acima, poderá usar a opção a seguir:

### **MQCNO\_NONE**

Nenhuma opção foi especificada

Use MQCNO\_NONE para auxiliar a documentação do programa. Não se pretende que essa opção seja usada com qualquer outra opção MQCNO\_\*, mas como seu valor é zero, tal uso não pode ser detectado.

### **ClientConnOffset (MQLONG) para MQCNO**

ClientConnOffset é o deslocamento em bytes de uma estrutura de definição de canal MQCD a partir do início da estrutura MQCNO. O deslocamento pode ser positivo ou negativo. Este campo é um campo de entrada com um valor inicial de 0.

Use *ClientConnOffset* apenas quando o aplicativo que emite a chamada MQCONNX estiver em execução como um IBM MQ MQI client. Para obter informações sobre como usar esse campo, consulte a descrição do campo *ClientConnPtr*.

Esse campo será ignorado se *Version* for menor que MQCNO\_VERSION\_2.

### **ClientConnPtr (MQPTR) para MQCNO**

ClientConnPtr é um campo de entrada. Seu valor inicial é o ponteiro nulo nessas linguagens de programação que suportam ponteiros e, caso contrário, uma cadeia de bytes all-null.

Use *ClientConnOffset* e *ClientConnPtr* apenas quando o aplicativo que emite a chamada MQCONNX estiver em execução como um IBM MQ MQI client. Especificando um ou outro desses campos, o aplicativo pode controlar a definição do canal de conexão do cliente fornecendo uma estrutura de definição de canal MQCD que contém os valores necessários.

Se o aplicativo estiver em execução como IBM MQ MQI client, mas não fornecer uma estrutura MQCD, a variável de ambiente MQSERVER será usada para selecionar a definição de canal. Se MQSERVER não for configurado, a tabela de canais do cliente será usada...

Se o aplicativo não estiver em execução como um IBM MQ MQI client, *ClientConnOffset* e *ClientConnPtr* serão ignoradas.

Se o aplicativo fornecer uma estrutura MQCD, configure os campos listados para os valores necessários; outros campos no MQCD serão ignorados.. É possível preencher sequências de caracteres com espaços em branco até o comprimento do campo ou finalizá-las com um caractere nulo. Consulte ["Campos"](#) na página 1533 para obter mais informações sobre os campos na estrutura MQCD..

Tabela 481. Campos em MQCD

<b>Campo em MQCD</b>	<b>Value</b>
<i>ChannelName</i>	Nome do canal.
<i>Version</i>	Número de versão da estrutura. Não deve ser menor que MQCD_VERSION_7..
<i>TransportType</i>	Qualquer tipo de transporte suportado..
<i>ModeName</i>	Nome do modo da LU 6.2
<i>TpName</i>	nome do programa de transação LU 6.2 .
<i>SecurityExit</i>	Nome da saída de segurança do canal.
<i>SendExit</i>	Nome da saída de envio do canal.
<i>ReceiveExit</i>	Nome da saída de recebimento do canal.
<i>MaxMsgLength</i>	Comprimento máximo em bytes de mensagens que podem ser enviadas pelo canal de conexão do cliente.
<i>SecurityUserData</i>	Dados do usuário para a saída de segurança
<i>SendUserData</i>	Dados do usuário para a saída de envio
<i>ReceiveUserData</i>	Dados do usuário para a saída de recepção

Tabela 481. Campos em MQCD (continuação)

Campo em MQCD	Value
<i>UserIdentifier</i>	Identificador de usuário a ser utilizado para estabelecer uma sessão LU 6.2 .
<i>Password</i>	Senha a ser utilizada para estabelecer uma sessão LU 6.2 .
<i>ConnectionName</i>	Nome da conexão.
<i>HeartbeatInterval</i>	Tempo em segundos entre fluxos de pulsação.
<i>StrucLength</i>	Comprimento da estrutura do MQCD
<i>ExitNameLength</i>	Comprimento de nomes de saída endereçados por <i>SendExitPtr</i> e <i>ReceiveExitPtr</i> Deve ser maior que zero se <i>SendExitPtr</i> ou <i>ReceiveExitPtr</i> for configurado para um valor que não seja o ponteiro nulo..
<i>ExitDataLength</i>	Comprimento dos dados de saída endereçados por <i>SendUserDataPtr</i> e <i>ReceiveUserDataPtr</i> .. Deve ser maior que zero se <i>SendUserDataPtr</i> ou <i>ReceiveUserDataPtr</i> for configurado para um valor que não seja o ponteiro nulo..
<i>SendExitsDefined</i>	Número de saídas de envio endereçadas por <i>SendExitPtr</i> .. Se zero, <i>SendExit</i> e <i>SendUserData</i> fornecerão o nome da saída e os dados. Se maior que zero, <i>SendExitPtr</i> e <i>SendUserDataPtr</i> fornecem os nomes e dados de saída e <i>SendExit</i> e <i>SendUserData</i> devem estar em branco.
<i>ReceiveExitsDefined</i>	Número de saídas de recebimento endereçadas por <i>ReceiveExitPtr</i> .. Se zero, <i>ReceiveExit</i> e <i>ReceiveUserData</i> fornecerão o nome da saída e os dados. Se maior que zero, <i>ReceiveExitPtr</i> e <i>ReceiveUserDataPtr</i> fornecem os nomes e dados de saída e <i>ReceiveExit</i> e <i>ReceiveUserData</i> devem estar em branco.
<i>SendExitPtr</i>	Endereço do nome da primeira saída de envio..
<i>SendUserDataPtr</i>	Endereço de dados para a primeira saída de envio
<i>ReceiveExitPtr</i>	Endereço do nome da primeira saída recebida.
<i>ReceiveUserDataPtr</i>	Endereço de dados para a primeira saída de recebimento
<i>LongRemoteUserIdLength</i>	Comprimento do identificador de usuário remoto longo.
<i>LongRemoteUserIdPtr</i>	Endereço do identificador de usuário remoto longo
<i>RemoteSecurityId</i>	Identificador de segurança remota.
<i>SSLCipherSpec</i>	TLS CipherSpec..
<i>SSLPeerNamePtr</i>	Endereço do nome do peer do TLS
<i>SSLPeerNameLength</i>	Comprimento do nome do peer do TLS
<i>KeepAliveInterval</i>	Valor transmitido para a pilha de comunicações para sincronização keep-alive para o canal
<i>LocalAddress</i>	O endereço de comunicações local, incluindo o endereço IP do adaptador de rede local a ser usado e um intervalo de portas a ser usado para conexões de saída..

Forneça a estrutura de definição de canal de uma de duas maneiras:.

- Usando o campo de deslocamento *ClientConnOffset*

Nesse caso, o aplicativo deve declarar uma estrutura composta contendo um MQCNO seguido pela estrutura de definição de canal MQCD e configurar *ClientConnOffset* para o deslocamento da

estrutura de definição do canal do início do MQCNO. Assegure que esse deslocamento esteja correto. *ClientConnPtr* deve ser configurado para o ponteiro nulo ou bytes nulos.

Use *ClientConnOffset* para linguagens de programação que não suportem o tipo de dados do ponteiro ou que implementem o tipo de dados do ponteiro de uma maneira que não seja móvel para diferentes ambientes (por exemplo, a linguagem de programação COBOL)..

Para a linguagem de programação Visual Basic, uma estrutura composta chamada MQCNOCD é fornecido no arquivo de cabeçalho CMQXB.BAS; essa estrutura contém uma estrutura MQCNO seguida por uma estrutura MQCD.. Inicialize o MQCNOCD chamando a sub-rotina MQCNOCD\_DEFAULTS. MQCNOCD é usado com o MQCONNXAny variante da chamada MQCONNX; consulte a descrição da chamada MQCONNX para obter detalhes adicionais..

- Usando o campo de ponteiro *ClientConnPtr* .

Neste caso, o aplicativo pode declarar a estrutura de definição de canal separadamente da estrutura MQCNO e configurar *ClientConnPtr* para o endereço da estrutura de definição de canal.. Configure *ClientConnOffset* como zero..

Use *ClientConnPtr* para linguagens de programação que suportem o tipo de dados do ponteiro de uma maneira que seja móvel para ambientes diferentes (por exemplo, a linguagem de programação C)

Na linguagem de programação C, é possível usar a variável de macro MQCD\_CLIENT\_CONN\_DEFAULT para fornecer valores iniciais para a estrutura mais adequada para uso na chamada MQCONNX do que os valores iniciais fornecidos por MQCD\_DEFAULT.

Independentemente da técnica escolhida, é possível usar apenas um dos *ClientConnOffset* e *ClientConnPtr* ; a chamada falha com código de razão MQRC\_CLIENT\_CONN\_ERROR se ambos forem diferentes de zero.

Quando a chamada MQCONNX for concluída, a estrutura MQCD não será referenciada novamente.

Esse campo será ignorado se *Version* for menor que MQCNO\_VERSION\_2.

**Nota:** Em plataformas em que a linguagem de programação não suporta o tipo de dados do ponteiro, este campo é declarado como uma cadeia de bytes do comprimento apropriado, sendo o valor inicial a cadeia de bytes all-null.

### **ConnTag (MQBYTE128) para MQCNO em Multiplataformas**

Uma tag de conexão é conceitualmente semelhante a um identificador de conexão, mas pode abranger várias conexões relacionadas, identificando-as como uma única instância do aplicativo... Em Multiplataformas, a tag de conexão é gerada pelo Gerenciador de Filas no momento da conexão

Para obter mais informações, consulte [identificador de conexão](#) e [instância do aplicativo](#).

As tags de conexão geradas são semi legíveis. Ou seja, eles podem ser exibidos e filtrados no MQSC como se fossem sequências no conjunto de caracteres local. As conexões que são conhecidas pelo IBM MQ para serem relacionadas são designadas automaticamente à mesma tag de conexão. Essa designação é particularmente importante para o [balanceamento de aplicativo](#)

A tag de conexão gerada é visível de três maneiras:

- Na estrutura MQCNO de saída em uma chamada MQCONNX, quando [MQCNO\\_GENERATE\\_CONN\\_TAG](#) é especificado..
- Na saída de [DISPLAY CONN](#) (ou equivalentes programáticos).
- Na saída de [DISPLAY APSTATUS](#) (ou equivalentes).

A tag deixa de ser válida quando o aplicativo termina ou emite a chamada MQDISC.

### **Referências relacionadas**

[“ConnTag \(MQBYTE128\) para MQCNO no IBM MQ for z/OS.”](#) na página 340

Uma tag de conexão é conceitualmente semelhante a um identificador de conexão, mas pode abranger várias conexões relacionadas, identificando-as como uma única instância do aplicativo.. No IBM MQ for z/OS, a tag de conexão é um campo de entrada, fornecido pelo aplicativo e usado em conjunto com as opções MQCNO\_\*\_CONN\_TAG para serializar conexões a partir dessa instância do aplicativo

## **ConnTag (MQBYTE128) para MQCNO no IBM MQ for z/OS .**

Uma tag de conexão é conceitualmente semelhante a um identificador de conexão, mas pode abranger várias conexões relacionadas, identificando-as como uma única instância do aplicativo.. No IBM MQ for z/OS, a tag de conexão é um campo de entrada, fornecido pelo aplicativo e usado em conjunto com as opções MQCNO\_\*\_CONN\_TAG para serializar conexões a partir dessa instância do aplicativo

Onde houver várias instâncias de um aplicativo que devem ser conectadas simultaneamente, cada uma delas deverá fornecer um valor exclusivo para esse campo.. Consulte as descrições dessas opções de tag de conexão para obter detalhes adicionais

### **Notas:**

- No IBM MQ for z/OS, não há como determinar administrativamente a tag de conexão associada a um aplicativo no tempo de execução.
- Os valores de tag de conexão que começam com MQ em maiúsculas, minúsculas ou letras maiúsculas e minúsculas em ASCII ou EBCDIC são reservados para uso pelos produtos IBM . Não use valores de tag de conexão começando com essas letras.

Use o valor especial a seguir se você não precisar de tag:


### **MQCT\_NONE**

O valor é binário zero para o comprimento do campo.

Para a linguagem de programação C, a constante MQCT\_NONE\_ARRAY também é definida; essa constante tem o mesmo valor que MQCT\_NONE, mas é uma matriz de caracteres em vez de uma cadeia.

O campo ConnTag é usado ao se conectar a um gerenciador de filas z/OS .

O comprimento desse campo é fornecido por MQ\_CONN\_TAG\_LENGTH. Esse campo será ignorado se *Version* for menor que MQCNO\_VERSION\_3.

 Consulte [“ConnTag \(MQBYTE128\) para MQCNO em Multiplataformas”](#) na página 339 para obter informações sobre como usar a tag de conexão em IBM MQ for Multiplatforms

## **SSLConfigPtr (PMQSCO) para MQCNO**

SSLConfigPtr é um campo de entrada. Seu valor inicial é o ponteiro nulo nessas linguagens de programação que suportam ponteiros e, caso contrário, uma cadeia de bytes all-null.

Use *SSLConfigPtr* e *SSLConfigOffset* apenas quando o aplicativo que emite a chamada MQCONNX estiver em execução como um IBM MQ MQI cliente e o protocolo de canal for TCP/IP. Se o aplicativo não estiver em execução como um cliente IBM MQ ou o protocolo do canal não for TCP/IP, *SSLConfigPtr* e *SSLConfigOffset* serão ignorados.

Especificando *SSLConfigPtr* ou *SSLConfigOffset*, mais *ClientConnPtr* ou *ClientConnOffset*, o aplicativo pode controlar o uso de TLS para a conexão do cliente. Quando as informações do TLS são especificadas dessa maneira, as variáveis de ambiente MQSSLKEYR e MQSSLCRYP são ignoradas; quaisquer informações relacionadas ao TLS na tabela de definições de canal do cliente (CCDT) também são ignoradas.

As informações de TLS podem ser especificadas apenas em:

- A primeira chamada MQCONNX do processo do cliente ou
- Uma chamada MQCONNX subsequente quando todas as conexões TLS anteriores com o gerenciador de filas foram concluídas usando MQDISC.

Esses são os únicos estados nos quais o ambiente TLS do processo pode ser inicializado. Se uma chamada MQCONNX for emitida especificando informações de TLS quando o ambiente TLS já existir, as informações de TLS na chamada serão ignoradas e a conexão será feita usando o ambiente TLS existente; a chamada retornará o código de conclusão MQCC\_WARNING e o código de razão MQRC\_SSL\_ALREADY\_INITIALIZED neste caso.



É possível fornecer a estrutura MQSCO da mesma maneira que a estrutura MQCD, especificando um endereço em *SSLConfigPtr* ou especificando um deslocamento em *SSLConfigOffset*; consulte a descrição de *ClientConnPtr* para obter detalhes de como fazer isso. No entanto, não é possível usar mais de um de *SSLConfigPtr* e *SSLConfigOffset*; a chamada falha com código de razão MQRC\_SSL\_CONFIG\_ERROR, se ambos forem diferentes de zero.

Quando a chamada MQCONNX for concluída, a estrutura MQSCO não será referenciada novamente.

Esse campo será ignorado se o *Version* for menor que MQCNO\_VERSION\_4

**Nota:** Nas plataformas em que a linguagem de programação não suporta o tipo de dados do ponteiro, este campo é declarado como uma cadeia de bytes do comprimento apropriado.

### ***SSLConfigOffset (MQLONG) para MQCNO***

*SSLConfigOffset* é o deslocamento em bytes de uma estrutura MQSCO do início da estrutura MQCNO. O deslocamento pode ser positivo ou negativo. Este campo é um campo de entrada, com um valor inicial de 0.

Use *SSLConfigOffset* apenas quando o aplicativo que emite a chamada MQCONNX estiver em execução como um IBM MQ MQI client. Para obter informações sobre como usar esse campo, consulte a descrição do campo *SSLConfigPtr*.

Esse campo será ignorado se o *Version* for menor que MQCNO\_VERSION\_4

### ***ConnectionId (MQBYTE24) para MQCNO***

*ConnectionId* é um identificador exclusivo de 24 bytes que permite IBM MQ identificar de forma confiável um aplicativo. Um aplicativo pode usar esse identificador para correlação em chamadas PUT e GET. Este parâmetro de saída tem um valor inicial de 24 bytes nulos em todas as linguagens.

O gerenciador de filas designa um ID exclusivo para todas as conexões, entretanto, elas são estabelecidas. Se um MQCONNX estabelecer a conexão com um MQCNO versão 5, o aplicativo poderá determinar o *ConnectionId* do MQCNO retornado. O identificador designado é garantido ser exclusivo entre todos os outros identificadores que o IBM MQ gera, como *CorrelId*, *MsgID* e *GroupId*.

Use *ConnectionId* para identificar unidades de trabalho de longa execução usando o comando PCF Inquire Connection ou o comando MQSC DISPLAY CONN. O *ConnectionId* usado pelos comandos MQSC (CONN) é derivado do *ConnectionId* retornado aqui. Os comandos PCF Inquire e Stop Connection podem utilizar o *ConnectionId* retornado aqui sem modificação.

É possível usar o *ConnectionId* para forçar o término de uma unidade de trabalho de longa execução, especificando o *ConnectionId* usando o comando PCF Stop Connection ou o comando MQSC STOP CONN. Consulte [Parar Conexão](#) e [STOP CONN](#) para obter mais informações sobre como usar esses comandos.

Esse campo não será retornado se a Versão for menor do que MQCNO\_VERSION\_5

O comprimento desse campo é fornecido por MQ\_CONNECTION\_ID\_LENGTH

### ***SecurityParmsDeslocamento (MQLONG) para MQCNO***

*SecurityParmsOffset* é o deslocamento em bytes da estrutura MQCSP a partir do início da estrutura MQCNO. O deslocamento pode ser positivo ou negativo. Este campo é um campo de entrada, com um valor inicial de 0.

Esse campo será ignorado se a *Version* for menor que MQCNO\_VERSION\_5

A estrutura do MQCSP é definida em [“MQCSP-Parâmetros de segurança”](#) na página 343

### ***SecurityParmsPtr (PMQCSP) para MQCNO***

*SecurityParmsPtr* é o endereço da estrutura MQCSP, usado para especificar um ID do usuário e senha para autenticação pelo serviço de autorização. Esse campo é um campo de entrada e seu valor inicial é um ponteiro nulo ou bytes nulos.

Esse campo será ignorado se a *Version* for menor que MQCNO\_VERSION\_5

A estrutura do MQCSP é definida em [“MQCSP-Parâmetros de segurança” na página 343](#)

### **Reservado (MQBYTE4) para MQCNO**

Um campo reservado para preencher a estrutura para um limite de 64 bits. O valor inicial do campo é zero binário para o comprimento do campo.

Esse campo será ignorado se o *Version* for menor que MQCNO\_VERSION\_6

### **CCDTUrlLength (MQLONG) para MQCNO**

CCDTUrlLength é o comprimento da sequência identificada por CCDTUrlPtr ou CCDTUrlOffset que contém uma URL que identifica o local da tabela do canal de conexão do cliente a ser usado para a conexão. O valor inicial do campo é zero.

Use CCDTUrlLength apenas quando o aplicativo que emite a chamada MQCONNX estiver em execução como um IBM MQ MQI client.

Esta é uma alternativa programática para configurar as variáveis de ambiente [MQCHLLIB](#) e [MQCHLTA](#).

Se o aplicativo não estiver executando como um cliente, CCDTUrlLength será ignorado.

Esse campo será ignorado se o *Version* for menor que MQCNO\_VERSION\_6

### **CCDTUrlPtr (PMQCHAR) para MQCNO**

CCDTUrlPtr é um ponteiro opcional para uma sequência que contém uma URL, para identificar o local da tabela de canal de conexão do cliente para usar para a conexão. Este campo é um campo de entrada, com um valor inicial de um ponteiro nulo em linguagens de programação que suportam ponteiros e, caso contrário, uma cadeia de bytes all-null.

Use CCDTUrlPtr apenas quando o aplicativo que emite a chamada MQCONNX estiver em execução como um IBM MQ MQI client.

**Importante:** É possível usar apenas um dos CCDTUrlPtr e CCDTUrlOffset. A chamada falha com o código de razão MQRC\_CCDT\_URL\_ERROR se ambos os campos não forem zero.

Esta é uma alternativa programática para configurar as variáveis de ambiente [MQCHLLIB](#) e [MQCHLTA](#).

Se o aplicativo não estiver executando como um cliente, CCDTUrlPtr será ignorado.

Esse campo será ignorado se o *Version* for menor que MQCNO\_VERSION\_6

### **CCDTUrlOffset (MQLONG) para MQCNO**

CCDTUrlOffset é o deslocamento em bytes, do início da estrutura MQCNO, para uma sequência que contém uma URL que identifica o local da tabela de canal de conexão do cliente a ser usado para a conexão. A compensação pode ser positiva ou negativa e o valor inicial do campo é zero.

Use CCDTUrlOffset apenas quando o aplicativo que emite a chamada MQCONNX estiver em execução como um IBM MQ MQI client.

**Importante:** É possível usar apenas um dos CCDTUrlPtr e CCDTUrlOffset. A chamada falha com o código de razão MQRC\_CCDT\_URL\_ERROR se ambos os campos não forem zero.

Esta é uma alternativa programática para configurar as variáveis de ambiente [MQCHLLIB](#) e [MQCHLTA](#).

Se o aplicativo não estiver executando como um cliente, CCDTUrlOffset será ignorado.

Esse campo será ignorado se o *Version* for menor que MQCNO\_VERSION\_6

### **AppName (MQCHAR28) para MQCNO**

O nome configurado pelo aplicativo para identificar a conexão com o gerenciador de filas. O valor inicial do campo é MQAN\_NONE\_ARRAY (caracteres em branco).

Esse campo será ignorado se *Version* for menor que MQCNO\_VERSION\_7 ou se o valor for configurado como espaços em branco.

**z/OS** Não é possível configurar esse campo em z/OS. Se você tentar fazer isso, receberá o código de razão MQR\_CNO\_ERROR novamente.

### **Reserved2 (MQBYTE4) para MQCNO**

Um campo reservado para preencher a estrutura para um limite de 64 bits. O valor inicial do campo é zero binário para o comprimento do campo.

Esse campo será ignorado se *Version* for menor que MQCNO\_VERSION\_7..

### **V 9.3.0 BalanceParmsOffset (MQLONG) para MQCNO**

O local da memória para uma estrutura do tipo MQBNO que contém informações sobre o comportamento de balanceamento do aplicativo. A estrutura é ignorada inteiramente, a menos que o aplicativo esteja se conectando por um canal do cliente.

Esse campo será ignorado se *Version* for menor que MQCNO\_VERSION\_8..

Consulte [MQBNO](#) para obter mais informações..

Se você fornecer esse campo, não será possível fornecer o campo “[BalanceParmsPtr \(MQPTR\) para MQCNO](#)” na página 343. Se você tentar fornecer ambos os campos, você receberá um MQR\_CNO\_ERROR. Como esse campo é relevante apenas para conexões do cliente, o fornecimento desse campo em qualquer outro tipo de conexão também resulta em MQR\_CNO\_ERROR.

### **V 9.3.0 BalanceParmsPtr (MQPTR) para MQCNO**

Ponteiro para o local da memória para uma estrutura do tipo MQBNO que contém informações sobre o comportamento de balanceamento do aplicativo. A estrutura é ignorada inteiramente, a menos que o aplicativo esteja se conectando por um canal do cliente.

Esse campo será ignorado se *Version* for menor que MQCNO\_VERSION\_8..

Consulte [MQBNO](#) para obter mais informações..

Se você fornecer esse campo, não será possível fornecer o campo “[BalanceParmsOffset \(MQLONG\) para MQCNO](#)” na página 343. Se você tentar fornecer ambos os campos, você receberá um MQR\_CNO\_ERROR. Como esse campo é relevante apenas para conexões do cliente, o fornecimento desse campo em qualquer outro tipo de conexão também resulta em MQR\_CNO\_ERROR.

## **MQCSP-Parâmetros de segurança**

A estrutura do parâmetro de segurança de conexão IBM MQ é usada por aplicativos para fluir informações de autenticação em uma chamada MQCONN para o gerenciador de filas. Também é possível usá-lo para fornecer a chave inicial usada com o sistema de proteção de senha do IBM MQ que criptografa dados sensíveis.

Configure *AuthenticationType* como MQCSP\_AUTH\_USER\_ID\_AND\_PWD para incluir o ID do usuário e a senha da versão 1.

Ao fornecer informações chave iniciais na versão 2, o *AuthenticationType* é padronizado para MQCSP\_AUTH\_NONE.

**V 9.3.4** De IBM MQ 9.3.4, use *AuthenticationType* para incluir informações do token de autenticação.

**V 9.3.4** É possível usar MQCSP\_AUTH\_USER\_ID\_AND\_PWD ou MQCSP\_AUTH\_ID\_TOKEN, mas não ambos.

**Aviso:** Em alguns casos, a senha ou o token de autenticação em uma estrutura MQCSP para um aplicativo cliente é enviado pela rede em texto simples. Para assegurar que as senhas do aplicativo cliente e os tokens de autenticação sejam protegidos apropriadamente, consulte [Proteção de senha do MQCSP](#).

## Disponibilidade

A estrutura MQCSP está disponível em todas as plataformas IBM MQ suportadas.

## Versão

Os arquivos de cabeçalho, COPY e INCLUDE fornecidos para as linguagens de programação suportadas contêm a versão mais recente de MQCSP, mas com o valor inicial do campo *Version* configurado como MQCSP\_VERSION\_1. Para usar os campos que não estão presentes na estrutura version-1, o aplicativo deve configurar o campo *Version* para o número da versão necessário.

## Conjunto de caracteres e codificação

Os dados no MQCSP devem estar no conjunto de caracteres e na codificação do gerenciador de filas locais, eles são fornecidos pelo atributo do gerenciador de filas do **CodedCharSetId** e MQENC\_NATIVE, respectivamente

## Campos

**Nota:** Na tabela a seguir, os campos são agrupados por uso e não alfabeticamente. Os tópicos filhos seguem a mesma sequência.



Nome e descrição do campo	Nome da constante	Valor inicial (se houver) da constante
<u>StrucId</u> (identificador de estrutura)	MQCSP_STRUC_ID	'CSP-'
<u>Versão</u> (número da versão da estrutura).	MQCSP_VERSION_1	1
<u>AuthenticationType</u> (tipo de autenticação)	Nenhum	MQCSP_AUTH_NONE
<u>Reserved1</u> (necessário para alinhamento do ponteiro em IBM i)	Nenhum	Cadeia nula ou espaços em branco
<u>CSPUserIdPtr</u> (endereço do ID do usuário).	Nenhum	Ponteiro nulo ou bytes nulos
<u>CSPUserIdOffset</u> (deslocamento do ID do usuário)	Nenhum	0
<u>CSPUserIdComprimento</u> (comprimento do ID do usuário)	Nenhum	0
<u>Reserved2</u> (necessário para alinhamento do ponteiro em IBM i)	Nenhum	Cadeia nula ou espaços em branco
<u>CSPPasswordPtr</u> (endereço de senha)	Nenhum	Ponteiro nulo ou bytes nulos
<u>CSPPasswordOffset</u> (deslocamento de senha)	Nenhum	0
<u>CSPPasswordLength</u> (comprimento da senha)	Nenhum	0
<b>Nota:</b> Os campos restantes serão ignorados se <i>Version</i> for menor que MQCSP_VERSION_2..		
 <u>Reserved3</u> (necessário para alinhamento do ponteiro em IBM i)	Nenhum	Cadeia nula ou espaços em branco
 <u>InitialKeyPtr</u>	Nenhum	Ponteiro nulo ou espaços em branco

Tabela 482. Campos em MQCSP para MQCS (continuação)

Nome e descrição do campo	Nome da constante	Valor inicial (se houver) da constante
<b>V 9.3.0</b> <b>V 9.3.0</b> <u>InitialKeyOffset</u> (deslocamento de chave inicial para o sistema de proteção de senha)	Nenhum	0
<b>V 9.3.0</b> <b>V 9.3.0</b> <u>InitialKeyComprimento</u> (comprimento da chave inicial para o sistema de proteção de senha)	Nenhum	0
<b>Nota:</b> Os campos restantes serão ignorados se <i>Version</i> for menor que MQCSP_VERSION_3..		
<b>V 9.3.4</b> <u>Reserved4</u> (necessário para alinhamento do ponteiro em IBM i)	Nenhum	Cadeia nula ou espaços em branco
<b>V 9.3.4</b> <u>TokenPtr</u> (endereço do token de autenticação)	Nenhum	Ponteiro nulo ou espaços em branco
<b>V 9.3.4</b> <u>TokenKeyOffset</u> (deslocamento do token de autenticação)	Nenhum	0
<b>V 9.3.4</b> <u>TokenLength</u> (comprimento do token de autenticação)	Nenhum	0
<b>Notas:</b> <ol style="list-style-type: none"> <li>1. O símbolo ~ representa um único caractere em branco.</li> <li>2. Na linguagem de programação C, a variável macro MQCSP_DEFAULT contém os valores que estão listados na tabela. Ele pode ser usado da seguinte maneira para fornecer valores iniciais para os campos na estrutura: <pre>MQCSP MyCSP = {MQCSP_DEFAULT};</pre> </li> </ol>		

## Declarações de idiomas

### Declaração C para MQCSP

```
V 9.3.0 V 9.3.0
typedef struct tagMQCSP MQCSP;
struct tagMQCSP {
    MQCHAR4    StrucId;           /* Structure identifier */
    MQLONG     Version;          /* Structure version number */
    MQLONG     AuthenticationType; /* Type of authentication */
    MQBYTE4    Reserved1;        /* Required for IBM i pointer alignment */
    MQPTR      CSPUserIdPtr;      /* Address of user ID */
    MQLONG     CSPUserIdOffset;   /* Offset of user ID */
    MQLONG     CSPUserIdLength;   /* Length of user ID */
    MQBYTE8    Reserved2;        /* Required for IBM i pointer alignment */
    MQPTR      CSPPasswordPtr;    /* Address of password */
    MQLONG     CSPPasswordOffset; /* Offset of password */
    MQLONG     CSPPasswordLength; /* Length of password */
    /* Ver:1 */

    MQBYTE8    Reserved3;        /* Required for IBM i pointer alignment */
    MQPTR      InitialKeyPtr;     /* Address of initial key */
    MQLONG     InitialKeyOffset;  /* Offset of initial key */
    MQLONG     InitialKeyLength;  /* Length of initial key */
    /* Ver:2 */
};
```

#### V 9.3.4

```
typedef struct tagMQCSP MQCSP;
struct tagMQCSP {
    MQCHAR4   StructId;           /* Structure identifier */
    MQLONG    Version;           /* Structure version number */
    MQLONG    AuthenticationType; /* Type of authentication */
    MQBYTE4   Reserved1;        /* Required for IBM i pointer alignment */
    MQPTR     CSPUserIdPtr;      /* Address of user ID */
    MQLONG    CSPUserIdOffset;   /* Offset of user ID */
    MQLONG    CSPUserIdLength;   /* Length of user ID */
    MQBYTE8   Reserved2;        /* Required for IBM i pointer alignment */
    MQPTR     CSPPasswordPtr;    /* Address of password */
    MQLONG    CSPPasswordOffset; /* Offset of password */
    MQLONG    CSPPasswordLength; /* Length of password */
/* Ver:1 */

    MQBYTE8   Reserved3;        /* Required for IBM i pointer alignment */
    MQPTR     InitialKeyPtr;     /* Address of initial key */
    MQLONG    InitialKeyOffset;  /* Offset of initial key */
    MQLONG    InitialKeyLength;  /* Length of initial key */
/* Ver:2 */

    MQBYTE8   Reserved4;        /* Required for IBM i pointer alignment */
    MQPTR     TokenPtr;          /* Address of token */
    MQLONG    TokenOffset;       /* Offset of token */
    MQLONG    TokenLength;       /* Length of token */
/* Ver:3 */
};
```

#### Declaração COBOL para MQCSP

#### V 9.3.0 V 9.3.0

```
** MQCSP structure
10 MQCSP.
** Structure identifier
15 MQCSP-STRUCID PIC X(4).
** Structure version number
15 MQCSP-VERSION PIC S9(9) BINARY.
** Type of authentication
15 MQCSP-AUTHENTICATIONTYPE PIC S9(9) BINARY.
** Required for IBM i pointer alignment
15 MQCSP-RESERVED1 PIC X(4).
** Address of user ID
15 MQCSP-CSPUSERIDPTR POINTER.
** Offset of user ID
15 MQCSP-CSPUSERIDOFFSET PIC S9(9) BINARY.
** Length of user ID
15 MQCSP-CSPUSERIDLENGTH PIC S9(9) BINARY.
** Required for IBM i pointer alignment
15 MQCSP-RESERVED2 PIC X(4).
** Address of password
15 MQCSP-CSPPASSWORDPTR POINTER.
** Offset of password
15 MQCSP-CSPPASSWORDOFFSET PIC S9(9) BINARY.
** Length of password
15 MQCSP-CSPPASSWORDLENGTH PIC S9(9) BINARY.
** Ver:1 **

** Reserved
15 MQCSP-RESERVED3 PIC X(8).
** Address of initial key
15 MQCSP-INITIALKEYPTR POINTER.
** Offset of initial key
15 MQCSP-INITIALKEYOFFSET PIC S9(9) BINARY.
** Length of initial key
15 MQCSP-INITIALKEYLENGTH PIC S9(9) BINARY.
** Ver:2 **
```

#### V 9.3.4

```
** MQCSP structure
10 MQCSP.
** Structure identifier
15 MQCSP-STRUCID PIC X(4).
** Structure version number
15 MQCSP-VERSION PIC S9(9) BINARY.
** Type of authentication
```

```

15 MQCSP-AUTHENTICATIONTYPE PIC S9(9) BINARY.
** Required for IBM i pointer alignment
15 MQCSP-RESERVED1          PIC X(4).
** Address of user ID
15 MQCSP-CSPUSERIDPTR      POINTER.
** Offset of user ID
15 MQCSP-CSPUSERIDOFFSET  PIC S9(9) BINARY.
** Length of user ID
15 MQCSP-CSPUSERIDLENGTH  PIC S9(9) BINARY.
** Required for IBM i pointer alignment
15 MQCSP-RESERVED2        PIC X(4).
** Address of password
15 MQCSP-CSPPASSWORDPTR   POINTER.
** Offset of password
15 MQCSP-CSPPASSWORDOFFSET PIC S9(9) BINARY.
** Length of password
15 MQCSP-CSPPASSWORDLENGTH PIC S9(9) BINARY.
** Ver:1 **

** Reserved
15 MQCSP-RESERVED3        PIC X(8).
** Address of initial key
15 MQCSP-INITIALKEYPTR    POINTER.
** Offset of initial key
15 MQCSP-INITIALKEYOFFSET PIC S9(9) BINARY.
** Length of initial key
15 MQCSP-INITIALKEYLENGTH PIC S9(9) BINARY.
** Ver:2 **

** Reserved
15 MQCSP-RESERVED4        PIC X(8).
** Address of token
15 MQCSP-TOKENPTR         POINTER.
** Offset of token
15 MQCSP-TOKENOFFSET     PIC S9(9) BINARY.
** Length of token
15 MQCSP-TOKENLENGTH     PIC S9(9) BINARY.
** Ver:3 **

```

## Declaração PL/I para MQCSP

```

V9.3.0 V9.3.0
dcl
1 MQCSP based,
3 StrucId          char(4),          /* Structure identifier */
3 Version          fixed bin(31),    /* Structure version number */
3 AuthenticationType fixed bin(31), /* Type of authentication */
3 Reserved1        char(4),          /* Required for IBM i pointer
                                     alignment */
3 CSPUserIdPtr     pointer,          /* Address of user ID */
3 CSPUserIdOffset  fixed bin(31),    /* Offset of user ID */
3 CSPUserIdLength  fixed bin(31),    /* Length of user ID */
3 Reserved2        char(8),          /* Required for IBM i pointer
                                     alignment */
3 CSPPasswordPtr   pointer,          /* Address of password */
3 CSPPasswordOffset fixed bin(31),    /* Offset of user ID */
3 CSPPasswordLength fixed bin(31), /* Length of user ID */
/* Version 1 */

3 Reserved3        char(8),          /* Reserved */
3 InitialKeyPtr    pointer,          /* Address of initial key */
3 InitialKeyOffset fixed bin(31),    /* Offset of initial key */
3 InitialKeyLength fixed bin(31); /* Length of initial key */
/* Version 2 */

```

```

V9.3.4
dcl
1 MQCSP based,
3 StrucId          char(4),          /* Structure identifier */
3 Version          fixed bin(31),    /* Structure version number */
3 AuthenticationType fixed bin(31), /* Type of authentication */
3 Reserved1        char(4),          /* Required for IBM i pointer
                                     alignment */
3 CSPUserIdPtr     pointer,          /* Address of user ID */
3 CSPUserIdOffset  fixed bin(31),    /* Offset of user ID */
3 CSPUserIdLength  fixed bin(31),    /* Length of user ID */
3 Reserved2        char(8),          /* Required for IBM i pointer
                                     alignment */

```



```

3 CSPPasswordPtr    pointer,      /* Address of password */
3 CSPPasswordOffset fixed bin(31), /* Offset of user ID */
3 CSPPasswordLength fixed bin(31), /* Length of user ID */
/* Version 1 */

3 Reserved3        char(8),      /* Reserved */
3 InitialKeyPtr    pointer,      /* Address of initial key */
3 InitialKeyOffset fixed bin(31), /* Offset of initial key */
3 InitialKeyLength fixed bin(31); /* Length of initial key */
/* Version 2 */

3 Reserved4        char(8),      /* Reserved */
3 TokenPtr         pointer,      /* Address of Token */
3 TokenOffset      fixed bin(31), /* Offset of Token */
3 TokenLength      fixed bin(31); /* Length of Token */
/* Version 3 */

```

## Declaração do Visual Basic para MQCSP

```

Type MQCSP
  StrucId          As String*4    'Structure identifier'
  Version          As Long        'Structure version number'
  AuthenticationType As Long      'Type of authentication'
  Reserved1        As MQBYTE4     'Required for IBM i pointer'
                                     'alignment'
  CSPUserIdPtr     As MQPTR       'Address of user ID'
  CSPUserIdOffset  As Long        'Offset of user ID'
  CSPUserIdLength  As Long        'Length of user ID'
  Reserved2        As MQBYTE8     'Required for IBM i pointer'
                                     'alignment'
  CSPPasswordPtr   As MQPTR       'Address of password'
  CSPPasswordOffset As Long      'Offset of password'
  CSPPasswordLength As Long      'Length of password'
End Type

```

## Conceitos relacionados

[Trabalhando com tokens de autenticação.](#)

### ***StrucId (MQCHAR4) para MQCSP***

Esse é o identificador da estrutura de parâmetros de segurança. Ele é sempre um campo de entrada Seu valor é MQCSP\_STRUC\_ID.

O valor deve ser:.

#### **MQCSP\_STRUC\_ID**

Identificador para a estrutura de parâmetros de segurança.

Para a linguagem de programação C, a constante MQCSP\_STRUC\_ID\_ARRAY também está definida. Ele tem o mesmo valor que MQCSP\_STRUC\_ID, mas é uma matriz de caracteres em vez de uma cadeia.

### ***Versão (MQLONG) para MQCSP***

Número da versão da estrutura do MQCSP

O valor deve ser:.

#### **MQCSP\_VERSION\_1**

Estrutura de parâmetros de segurança Version-1 . Na versão 1, é possível incluir um ID do usuário e senha na estrutura MQCSP para autenticar com o gerenciador de filas.

#### **V 9.3.0** **MQCSP\_VERSION\_2**

Estrutura de parâmetros de segurança Version-2 . Na versão 2, é possível incluir um ID do usuário e uma senha para autenticar com o gerenciador de filas e especificar a chave inicial que é usada para proteger as senhas.

#### **V 9.3.4** **MQCSP\_VERSION\_3**

Estrutura de parâmetros de segurança Version-3 . Na versão 3, é possível incluir um ID do usuário e senha ou um token de autenticação na estrutura MQCSP para autenticar com o gerenciador de filas. Também é possível especificar a chave inicial usada para proteger senhas.



A constante a seguir especifica o número da versão atual:

### **MQCSP\_CURRENT\_VERSION**

A versão atual da estrutura de parâmetros de segurança

Isto é sempre um campo de entrada. O valor inicial desse campo é MQCSP\_VERSION\_3.

### **AuthenticationType (MQLONG) para MQCSP**

AuthenticationType é um campo de entrada. Seu valor inicial é MQCSP\_AUTH\_NONE.

Esse é o tipo de autenticação a ser executado Os valores válidos são:

#### **MQCSP\_AUTH\_NONE**

Não use os campos ID do usuário e senha ou token de autenticação .

#### **MQCSP\_AUTH\_USER\_ID\_AND\_PWD**

Autentique usando o ID do usuário e a senha na estrutura MQCSP.

#### **V 9.3.4 MQCSP\_AUTH\_ID\_TOKEN**

Autentique usando o token de autenticação na estrutura MQCSP.

O valor padrão é MQCSP\_AUTH\_NONE. Com a configuração padrão nenhuma proteção de senha é feita.

Se você requerer autenticação, deverá configurar **MQCSP.AuthenticationType** para MQCSP\_AUTH\_USER\_ID\_AND\_PWD ou MQCSP\_AUTH\_ID\_TOKEN

Para obter mais informações, consulte [Proteção de senha do MQCSP](#).

### **Conceitos relacionados**

[Trabalhando com tokens de autenticação.](#)

### **Reserved1 (MQBYTE4) para MQCSP**

Um campo reservado, obrigatório para alinhamento do ponteiro em IBM i.

O valor inicial deste campo é nulo.

### **CSPUserIdPtr (MQPTR) para MQCSP**

O endereço para o ID do usuário a ser usado na autenticação

Este é um campo de entrada. O valor inicial deste campo é o ponteiro nulo nas linguagens de programação que suportam ponteiros e, caso contrário, uma cadeia de bytes all-null. Esse campo será ignorado se *Version* for menor que MQCNO\_VERSION\_5.

Este campo pode conter um ID do usuário do sistema operacional quando um **AUTHTYPE** de *IDPWOS* é nomeado no campo [CONNAUTH](#) do gerenciador de filas..

No Windows , pode ser um ID do usuário do domínio completo.

Este campo pode conter um ID do usuário LDAP quando um **AUTHTYPE** de *IDPWLDAP* é nomeado no campo [CONNAUTH](#) do gerenciador de filas.

### **CSPUserIdOffset (MQLONG) para MQCSP**

O deslocamento em bytes para o ID do usuário a ser usado na autenticação. O deslocamento pode ser positivo ou negativo

Este é um campo de entrada. O valor inicial desse campo é 0.

### **CSPUserIdComprimento (MQLONG) para MQCSP**

O comprimento do ID do usuário a ser usado na autenticação.

O comprimento máximo do ID do usuário depende da plataforma, consulte [IDs do usuário](#).

**V 9.3.0** **V 9.3.0** Se o comprimento do ID do usuário for maior que o comprimento máximo

permitido, a solicitação de autenticação falhará com MQRC\_CSP\_ERROR. Em versões anteriores do IBM MQo erro retornado é MQRC\_NOT\_AUTHORIZED.

Este campo é um campo de entrada O valor inicial desse campo é 0.

### **Reserved2 (MQBYTE8) para MQCSP**

Um campo reservado, obrigatório para alinhamento do ponteiro em IBM i.

O valor inicial deste campo é nulo.

### **CSPPasswordPtr (MQPTR) para MQCSP**

O endereço para a senha a ser usada na autenticação

Este é um campo de entrada. O valor inicial deste campo é o ponteiro nulo nas linguagens de programação que suportam ponteiros e, caso contrário, uma cadeia de bytes all-null. Esse campo será ignorado se *Version* for menor que MQCNO\_VERSION\_5.

Esse campo pode conter uma senha vazia que é rejeitada pelo sistema operacional ou pela verificação de senha LDAP, dependendo da configuração, mas não é rejeitada pelo IBM MQ antes de ser transmitida para o método de autenticação

### **CSPPasswordOffset (MQLONG) para MQCSP**



Este é o deslocamento em bytes para a senha a ser usada na autenticação. O deslocamento pode ser positivo ou negativo

Este é um campo de entrada. O valor inicial desse campo é 0.

### **CSPPasswordLength (MQLONG) para MQCSP**

O comprimento da senha a ser usada na autenticação

O comprimento máximo da senha é MQ\_CSP\_PASSWORD\_LENGTH, que é 256 caracteres.

  Se o comprimento da senha for maior do que o comprimento máximo permitido, a solicitação de autenticação falhará com MQRC\_CSP\_ERROR Em versões anteriores do IBM MQo erro retornado é MQRC\_NOT\_AUTHORIZED.

Este campo é um campo de entrada O valor inicial desse campo é 0.

### **Reserved3 (MQBYTE8) para MQCSP**

Um campo reservado, necessário para o alinhamento do ponteiro em IBM i

O valor inicial deste campo é nulo.

### **InitialKeyPtr (MQPTR) para MQCSP**

O endereço para a chave inicial para o sistema de proteção de senha

Este é um campo de entrada. O valor inicial deste campo é o ponteiro nulo nas linguagens de programação que suportam ponteiros e, caso contrário, uma cadeia de bytes all-null. Esse campo será ignorado se *Version* for menor que MQCSP\_VERSION\_2.

Esse campo é relevante apenas para IBM MQ MQI clients em execução em IBM i, AIX, Linux, and Windows sistemas....

IBM MQ MQI clients pode fornecer valores criptografados para alguns campos, usando o sistema de proteção de senha IBM MQ . Se você usou uma chave inicial para criptografar a senha para o repositório de chaves especificado na estrutura MQCSO, assegure-se de incluir campos de chave iniciais no MQCSP para o mesmo aplicativo cliente.

Os clientes MQI do IBM MQ podem fornecer valores criptografados para alguns campos usando o sistema de proteção de senha IBM MQ . Se você usou uma chave inicial para criptografar a senha para o

repositório de chaves especificado na estrutura MQCSO, assegure-se de incluir campos de chave iniciais na estrutura MQCSP para o mesmo aplicativo cliente.

Uma chave inicial é usada pelo algoritmo de criptografia para criptografar e descriptografar esses valores. Se uma chave inicial for fornecida quando os valores desses campos forem criptografados usando o utilitário **runmqicred**, então a mesma chave inicial deverá ser especificada pelo cliente quando ele se conectar ao gerenciador de fila

A chave inicial especificada usando esse campo substitui qualquer chave inicial especificada usando a variável de ambiente `MQS_MQI_KEYFILE` ou a propriedade `MQIInitialKeyFile` na sub-rotina Segurança do arquivo de configuração do cliente.

É possível usar `InitialKeyOffset` ou `InitialKeyPtr` para especificar a chave inicial, mas não ambos.

#### Tarefas relacionadas

[Fornecendo uma chave inicial para um cliente MQI do IBM MQ no AIX, Linux e Windows](#)

[Protegendo senhas nos arquivos de configuração do componente IBM MQ](#)

#### Referências relacionadas

[runmqicred \(proteger senhas do cliente IBM MQ\)](#)

[“KeyRepoPasswordPtr \(MQPTR\) para MQSCO” na página 587](#)

Este é o endereço em bytes da passphrase do repositório de chaves TLS

[“InitialKeyOffset \(MQLONG\) para MQCSP” na página 351](#)

O deslocamento em bytes para a chave inicial para o sistema de proteção de senha a partir do início da estrutura MQCSP. O deslocamento pode ser positivo ou negativo

#### **InitialKeyOffset (MQLONG) para MQCSP**

O deslocamento em bytes para a chave inicial para o sistema de proteção de senha a partir do início da estrutura MQCSP. O deslocamento pode ser positivo ou negativo

É possível usar `InitialKeyOffset` ou `InitialKeyPtr` para especificar a chave inicial, mas não ambos. Para obter mais informações, consulte a descrição do campo `InitialKeyPtr`.

Este é um campo de entrada. O valor inicial desse campo é 0. Esse campo será ignorado se `Version` for menor que `MQCSP_VERSION_2`.

#### Tarefas relacionadas

[Protegendo senhas nos arquivos de configuração do componente IBM MQ](#)

[Fornecendo uma chave inicial para um cliente MQI do IBM MQ no AIX, Linux e Windows](#)

#### Referências relacionadas

[“InitialKeyPtr \(MQPTR\) para MQCSP” na página 350](#)

O endereço para a chave inicial para o sistema de proteção de senha

#### **InitialKeyComprimento (MQLONG) para MQCSP**

O comprimento da chave inicial para o sistema de proteção de senha.

Este é um campo de entrada. O valor inicial desse campo é 0. Esse campo será ignorado se `Version` for menor que `MQCSP_VERSION_2`.

#### Tarefas relacionadas

[Protegendo senhas nos arquivos de configuração do componente IBM MQ](#)

[Fornecendo uma chave inicial para um cliente MQI do IBM MQ no AIX, Linux e Windows](#)

#### **Reserved4 (MQBYTE8) para MQCSP**

Um campo reservado, necessário para o alinhamento do ponteiro em IBM i

O valor inicial deste campo é nulo.

### **V 9.3.4** *TokenPtr (MQPTR) para MQCSP*

O endereço do token de autenticação usado para autenticação com o gerenciador de filas.

Este é um campo de entrada. O valor inicial desse campo é o ponteiro nulo nas linguagens de programação que suportam ponteiros e, caso contrário, uma cadeia de bytes all-null. Esse campo será ignorado se *Version* for menor que MQCSP\_VERSION\_3.

Esse campo é relevante para IBM MQ MQI clients conectar a IBM MQ gerenciadores de filas que são executados em sistemas AIX ou Linux .

É possível usar *TokenOffset* ou *TokenPtr* para especificar o token de autenticação, mas não ambos.

Para obter mais informações, consulte [Usando tokens de autenticação em um aplicativo](#)

#### **Conceitos relacionados**

[Trabalhando com tokens de autenticação.](#)

#### **Referências relacionadas**

“TokenOffset (MQLONG) para MQCSP” na página 352

Este é o deslocamento em bytes para o token de autenticação do início da estrutura MQCSP. O deslocamento pode ser positivo ou negativo

### **V 9.3.4** *TokenOffset (MQLONG) para MQCSP*

Este é o deslocamento em bytes para o token de autenticação do início da estrutura MQCSP. O deslocamento pode ser positivo ou negativo

Este é um campo de entrada. O valor inicial desse campo é 0. Esse campo será ignorado se *Version* for menor que MQCSP\_VERSION\_3.

É possível usar *TokenOffset* ou *TokenPtr* para especificar o token, mas não ambos. Para obter mais informações, consulte a descrição do campo *TokenPtr* ..

#### **Conceitos relacionados**

[Trabalhando com tokens de autenticação.](#)

#### **Tarefas relacionadas**

[Usando tokens de autenticação em um aplicativo](#)

#### **Referências relacionadas**

“TokenPtr (MQPTR) para MQCSP” na página 352

O endereço do token de autenticação usado para autenticação com o gerenciador de filas.

### **V 9.3.4** *TokenLength (MQLONG) para MQCSP*

Esse é o comprimento do token de autenticação usado para a autenticação com o gerenciador de fila

O comprimento máximo do token de autenticação é MQ\_CSP\_TOKEN\_LENGTH, que é de 8192 bytes Se *TokenLength* for maior que o comprimento máximo permitido, a solicitação de autenticação falhará com MQRC\_CSP\_ERROR.

Este é um campo de entrada. O valor inicial desse campo é 0. Esse campo será ignorado se *Version* for menor que MQCSP\_VERSION\_3.

Para obter mais informações, consulte [Usando tokens de autenticação em um aplicativo](#)

#### **Conceitos relacionados**

[Trabalhando com tokens de autenticação.](#)

## **MQCTLO-Estrutura de opções de retorno de chamada de controle**

A estrutura MQCTLO é usada para especificar opções relacionadas a uma função de retorno de chamadas de controle. A estrutura é um parâmetro de entrada e de saída na chamada MQCTL.

## Disponibilidade

A estrutura MQCTLO está disponível nas plataformas a seguir:

-  AIX
-  IBM i
-  Linux
-  Windows
-  z/OS

e para o IBM MQ MQI clients conectado a esses sistemas

## Versão

A versão atual de MQCTLO é MQCTLO\_VERSION\_1.

## Conjunto de caracteres e codificação

Os dados no MQCTLO devem estar no conjunto de caracteres fornecido pelo atributo do gerenciador de filas do **CodedCharSetId** e pela codificação do gerenciador de fila local fornecido pelo MQENC\_NATIVE. No entanto, se o aplicativo estiver em execução como um cliente MQI MQ, a estrutura deverá estar no conjunto de caracteres e na codificação do cliente.

## Campos

**Nota:** Na tabela a seguir, os campos são agrupados por uso e não alfabeticamente. Os tópicos filhos seguem a mesma sequência.

<i>Tabela 483. Campos em MQCTLO</i>		
Nome e descrição do campo	Nome da constante	Valor inicial (se houver) da constante
<u>StrucID</u> (identificador de estruturas).	MQCTLO_STRUC_ID	'CTLO'
<u>Versão</u> (número da versão da estrutura).	MQCTLO_VERSION_1	1
<u>Opções</u> (opções).	MQCTLO_NONE	Nulos
<u>Opções</u> (campo reservado)	Campo reservado	
<u>ConnectionArea</u> (campo para a função de retorno de chamada usar)	Nenhum	Ponteiro nulo ou bytes nulos
<b>Notas:</b>		
1. Na linguagem de programação C, a variável macroMQCTLO_DEFAULT contém os valores listados na tabela. Use-o da seguinte maneira para fornecer valores iniciais para os campos na estrutura:		
<pre>MQCTLO MyCTLO = {MQCTLO_DEFAULT};</pre>		

## Declarações de idiomas

Declaração C para MQCTLO

```
typedef struct tagMQCTLO MQCTLO;  
struct tagMQCTLO {  
    MQCHAR4 StrucId;        /* Structure identifier */
```

```

MQLONG  Version;          /* Structure version number */
MQLONG  Options;         /* Options that control the action of MQCTL */
MQLONG  Reserved;       /* Reserved field */

MQPTR   ConnectionArea; /* Connection work area passed to the function */
};

```

## Declaração COBOL para MQCTLO

```

** MQCTLO structure
10  MQCTLO.
** Structure Identifier
15  MQCTLO-STRUCID          PIC X(4).
** Structure Version
15  MQCTLO-VERSION        PIC S9(9) BINARY.
** Options
15  MQCTLO-OPTIONS        PIC S9(9) BINARY.
** Reserved
15  MQCTLO-RESERVED        PIC S9(9) BINARY.
** ConnectionArea
15  MQCTLO-CONNECTIONAREA  POINTER

```

## Declaração PL/I para MQCTLO

```

dcl
1  MQCTLO based,
3  StrucId          char(4),          /* Structure identifier */
3  Version          fixed bin(31),   /* Structure version */
3  Options          fixed bin(31),   /* Options */
3  Reserved         fixed bin(31),
3  ConnectionArea  pointer;         /* Connection work area */

```

### **StrucId (MQCHAR4) para MQCTLO**

Este é o identificador da estrutura de opções de controle. Ele é sempre um campo de entrada Seu valor é MQCTLO\_STRUC\_ID.

O valor deve ser:.

#### **MQCTLO\_STRUC\_ID**

Identificador para a estrutura de opções de controle.

Para a linguagem de programação C, a constante MQCTLO\_STRUC\_ID\_ARRAY também é definida. Ele tem o mesmo valor que MQCTLO\_STRUC\_ID, mas é uma matriz de caracteres em vez de uma sequência.

### **Versão (MQLONG) para MQCTLO**

Estrutura de Opções de Controle-Campo Versão

Este é o número da versão da estrutura; o valor deve ser:

#### **MQCTLO\_VERSION\_1**

Version-1 Estrutura de opções de controle.

A constante a seguir especifica o número da versão atual:

#### **MQCTLO\_CURRENT\_VERSION**

A versão atual da estrutura de opções de Controle.

Isto é sempre um campo de entrada. O valor inicial desse campo é MQCTLO\_VERSION\_1.

### **Opções (MQLONG) para MQCTLO**

Estrutura de Opções de Controle-Campo Opções

Opções que controlam a ação de MQCTL.

#### **MQCTLO\_FAIL\_IF QUIESCING**

Force a chamada MQCTL a falhar se o gerenciador de filas ou a conexão estiver no estado de quiesce

Especifique MQGMO\_FAIL\_IF QUIESCING, nas opções MQGMO transmitidas na chamada MQCB, para causar notificação aos consumidores de mensagens quando eles estiverem quiesce.

### **MQCTLO\_THREAD\_AFFINITY**

Essa opção informa o sistema de que o aplicativo requer que todos os consumidores de mensagem, para a mesma conexão, sejam chamados no mesmo encadeamento. Esse encadeamento será usado para todas as chamadas dos consumidores até que a conexão seja interrompida.

**Opção padrão:** se você não precisar de nenhuma das opções descritas, use a seguinte opção:

### **MQCTLO\_NONE**

Use esse valor para indicar que nenhuma outra opção foi especificada. Todas as opções assumem seus valores-padrão. MQCTLO\_NONE é definido para auxiliar a documentação do programa; não se pretende que essa opção seja usada com nenhuma outra, mas como seu valor é zero, tal uso não pode ser detectado.

Este é um campo de entrada. O valor inicial do campo *Options* é MQCTLO\_NONE.

### **Reservado (MQLONG) para MQCTLO**

Este é um campo reservado. O valor deve ser zero.

### **ConnectionArea (MQPTR) para MQCTLO**

Estrutura de opções de controle-campo ConnectionArea ..

Este é um campo que está disponível para a função de retorno de chamada usar.

O gerenciador de filas não toma decisões com base no conteúdo desse campo e é transmitido inalterado para o campo ConnectionArea na estrutura MQCBC, que é um parâmetro de entrada para o retorno de chamada.

Esse campo é ignorado para todas as operações diferentes de MQOP\_START e MQOP\_START\_WAIT..

Este é um campo de entrada e saída para a função de retorno de chamada. O valor inicial desse campo é um ponteiro nulo ou bytes nulos.

## **MQDH - Cabeçalho de distribuição**

A estrutura MQDH descreve os dados adicionais que estão presentes em uma mensagem quando essa é uma mensagem da lista de distribuição armazenada em uma fila de transmissão. Uma mensagem da lista de distribuições é uma mensagem enviada para várias filas de destino. Os dados adicionais consistem na estrutura MQDH seguida por uma matriz de registros MQOR e uma matriz de registros MQPMR..Essa estrutura é usada por aplicativos especializados que colocam mensagens diretamente em filas de transmissão ou que removem mensagens de filas de transmissão (por exemplo: agentes do canal de mensagem)... Os aplicativos que desejam colocar mensagens em listas de distribuição não devem usar essa estrutura. Em vez disso, eles devem usar a estrutura MQOD para definir os destinos na lista de distribuição e a estrutura MQPMO para especificar propriedades de mensagens ou receber informações sobre as mensagens enviadas aos destinos individuais.

## **Disponibilidade**

A estrutura MQDH está disponível nas plataformas a seguir:

-  AIX
-  IBM i
-  Linux
-  Windows

e para o IBM MQ MQI clients conectado a esses sistemas

## Nome do Formato

MQFMT\_DIST\_HEADER

## Conjunto de caracteres e codificação

Os dados em MQDH devem estar no conjunto de caracteres fornecido pelo atributo do gerenciador de filas do **CodedCharSetId** e pela codificação do gerenciador de filas locais fornecido por MQENC\_NATIVE

Configure o conjunto de caracteres e a codificação do MQDH nos campos *CodedCharSetId* e *Encoding* em:

- O MQMD (se a estrutura MQDH estiver no início dos dados da mensagem) ou
- A estrutura do cabeçalho que precede a estrutura MQDH (todos os outros casos).

## Uso

Quando um aplicativo coloca uma mensagem em uma lista de distribuição e alguns ou todos os destinos são remotos, o gerenciador de filas prefixa os dados da mensagem do aplicativo com as estruturas MQXQH e MQDH e coloca a mensagem na fila de transmissão relevante.. Portanto, os dados ocorrem na sequência a seguir quando a mensagem está em uma fila de transmissão:

- estrutura MQXQH
- Estrutura MQDH mais matrizes de registros MQOR e MQPMR
- Dados da mensagem do aplicativo

Dependendo dos destinos, o gerenciador de filas pode gerar mais de uma mensagem desse tipo e colocá-la em filas de transmissão diferentes. Nesse caso, as estruturas MQDH nessas mensagens identificam subconjuntos diferentes dos destinos definidos pela lista de distribuição aberta pelo aplicativo

Um aplicativo que coloca uma mensagem da lista de distribuição diretamente em uma fila de transmissão deve estar em conformidade com a sequência descrita anteriormente, e deve assegurar que a estrutura MQDH esteja correta. Se a estrutura MQDH não for válida, o gerenciador de fila poderá falhar a chamada MQPUT ou MQPUT1 com o código de razão MQRC\_DH\_ERROR..

Você pode armazenar mensagens em uma fila no formulário da lista de distribuição somente se tiver definido a fila como capaz de suportar mensagens da lista de distribuições. Consulte o atributo da fila **DistLists** descrito em “Atributos para filas” na página 865.. Se um aplicativo colocar uma mensagem da lista de distribuição diretamente em uma fila que não suporta listas de distribuição, o gerenciador de filas dividirá a mensagem da lista de distribuição em mensagens individuais e as colocará na fila.

## Campos

**Nota:** Na tabela a seguir, os campos são agrupados por uso e não alfabeticamente. Os tópicos filhos seguem a mesma sequência.

Nome e descrição do campo	Nome da constante	Valor inicial (se houver) da constante
<u>StrucId</u> (identificador de estrutura)	MQDH_STRUC_ID	'DH→→'
<u>Versão</u> (número da versão da estrutura).	MQDH_VERSION_1	1
<u>StrucLength</u> (comprimento da estrutura MQDH mais os seguintes registros)	Nenhum	0
<u>Codificação</u> (codificação numérica de dados que seguem matriz de registros MQPMR)	Nenhum	0



Tabela 484. Campos em MQDH para MQDH (continuação)

Nome e descrição do campo	Nome da constante	Valor inicial (se houver) da constante
<u>CodedCharSetId</u> (identificador do conjunto de caracteres de dados que segue a matriz dos registros MQPMR)	MQCCSI_UNDEFINED	0
<u>Formato</u> (nome do formato de dados que segue a matriz de registros MQPMR)..	MQFMT_NONE	Espaços em branco
<u>Flags</u> (sinalizadores gerais)	MQDHF_NONE	0
<u>PutMsgRecFields</u> (sinalizadores indicando quais campos MQPMR estão presentes)	MQPMRF_NONE	0
<u>RecsPresent</u> (número de registros de objeto presentes)	Nenhum	0
<u>Deslocamento doObjectRec</u> (deslocamento do primeiro registro de objeto do início do MQDH)	Nenhum	0
<u>PutMsgRecOffset</u> (deslocamento do primeiro registro put-message do início de MQDH)	Nenhum	0
<b>Notas:</b>		
<ol style="list-style-type: none"> <li>1. O símbolo ↵ representa um único caractere em branco.</li> <li>2. Na linguagem de programação C, a variável macroMQDH_DEFAULT contém os valores que são listados na tabela. Use-o da seguinte maneira para fornecer valores iniciais para os campos na estrutura: <pre>MQDH MyDH = {MQDH_DEFAULT};</pre> </li> </ol>		

## Declarações de idiomas

### Declaração C para MQDH

```
typedef struct tagMQDH MQDH;
struct tagMQDH {
    MQCHAR4  StrucId;          /* Structure identifier */
    MQLONG   Version;        /* Structure version number */
    MQLONG   StrucLength;    /* Length of MQDH structure plus following
                             MQOR and MQPMR records */
    MQLONG   Encoding;      /* Numeric encoding of data that follows
                             the MQOR and MQPMR records */
    MQLONG   CodedCharSetId; /* Character set identifier of data that
                             follows the MQOR and MQPMR records */
    MQCHAR8  Format;        /* Format name of data that follows the
                             MQOR and MQPMR records */
    MQLONG   Flags;        /* General flags */
    MQLONG   PutMsgRecFields; /* Flags indicating which MQPMR fields are
                             present */
    MQLONG   RecsPresent;   /* Number of MQOR records present */
    MQLONG   ObjectRecOffset; /* Offset of first MQOR record from start
                             of MQDH */
    MQLONG   PutMsgRecOffset; /* Offset of first MQPMR record from start
                             of MQDH */
};
```

### Declaração COBOL para MQDH

```
** MQDH structure
10 MQDH.
```

```

** Structure identifier
15 MQDH-STRUCID PIC X(4).
** Structure version number
15 MQDH-VERSION PIC S9(9) BINARY.
** Length of MQDH structure plus following MQOR and MQPMR records
15 MQDH-STRUCLength PIC S9(9) BINARY.
** Numeric encoding of data that follows the MQOR and MQPMR records
15 MQDH-ENCODING PIC S9(9) BINARY.
** Character set identifier of data that follows the MQOR and MQPMR
** records
15 MQDH-CODEDCHARSETID PIC S9(9) BINARY.
** Format name of data that follows the MQOR and MQPMR records
15 MQDH-FORMAT PIC X(8).
** General flags
15 MQDH-FLAGS PIC S9(9) BINARY.
** Flags indicating which MQPMR fields are present
15 MQDH-PUTMSGRECFIELDS PIC S9(9) BINARY.
** Number of MQOR records present
15 MQDH-RECPRESENT PIC S9(9) BINARY.
** Offset of first MQOR record from start of MQDH
15 MQDH-OBJECTRECOFFSET PIC S9(9) BINARY.
** Offset of first MQPMR record from start of MQDH
15 MQDH-PUTMSGRECOFFSET PIC S9(9) BINARY.

```

### Declaração PL/I para MQDH

```

dcl
1 MQDH based,
3 StrucId char(4), /* Structure identifier */
3 Version fixed bin(31), /* Structure version number */
3 StrucLength fixed bin(31), /* Length of MQDH structure plus
following MQOR and MQPMR
records */
3 Encoding fixed bin(31), /* Numeric encoding of data that
follows the MQOR and MQPMR
records */
3 CodedCharSetId fixed bin(31), /* Character set identifier of data
that follows the MQOR and MQPMR
records */
3 Format char(8), /* Format name of data that follows
the MQOR and MQPMR records */
3 Flags fixed bin(31), /* General flags */
3 PutMsgRecFields fixed bin(31), /* Flags indicating which MQPMR
fields are present */
3 RecsPresent fixed bin(31), /* Number of MQOR records present */
3 ObjectRecOffset fixed bin(31), /* Offset of first MQOR record from
start of MQDH */
3 PutMsgRecOffset fixed bin(31); /* Offset of first MQPMR record from
start of MQDH */

```

### Declaração do Visual Basic para MQDH.

```

Type MQDH
StrucId As String*4 'Structure identifier'
Version As Long 'Structure version number'
StrucLength As Long 'Length of MQDH structure plus following'
'MQOR and MQPMR records'
Encoding As Long 'Numeric encoding of data that follows'
'the MQOR and MQPMR records'
CodedCharSetId As Long 'Character set identifier of data that'
'follows the MQOR and MQPMR records'
Format As String*8 'Format name of data that follows the'
'MQOR and MQPMR records'
Flags As Long 'General flags'
PutMsgRecFields As Long 'Flags indicating which MQPMR fields are'
'present'
RecsPresent As Long 'Number of MQOR records present'
ObjectRecOffset As Long 'Offset of first MQOR record from start'
'of MQDH'
PutMsgRecOffset As Long 'Offset of first MQPMR record from start'
'of MQDH'
End Type

```

### ***StrucId (MQCHAR4) para MQDH***

Este é o identificador da estrutura do cabeçalho de distribuição. Ele é sempre um campo de entrada. Seu valor é MQDH\_STRUC\_ID.

O valor deve ser:.

#### **MQDH\_STRUC\_ID**

Identificador para a estrutura do cabeçalho de distribuição.

Para a linguagem de programação C, a constante MQDH\_STRUC\_ID\_ARRAY também é definida. Ele tem o mesmo valor que MQDH\_STRUC\_ID, mas é uma matriz de caracteres em vez de uma cadeia.

### ***Versão (MQLONG) para MQDH***

O valor deve ser:.

#### **MQDH\_VERSION\_1**

Número da versão para estrutura do cabeçalho de distribuição.

A constante a seguir especifica o número da versão atual:

#### **MQDH\_CURRENT\_VERSION**

Versão atual da estrutura do cabeçalho de distribuição..

O valor inicial desse campo é MQDH\_VERSION\_1.

### ***StrucLength (MQLONG) para MQDH***

Este é o número de bytes desde o início da estrutura MQDH até o início dos dados da mensagem após as matrizes de registros MQOR e MQPMR. Os dados ocorrem na seguinte sequência:

- Estrutura MQDH
- Matriz de registros MQOR
- Matriz de registros MQPMR
- Dados da mensagem

As matrizes de registros MQOR e MQPMR são tratadas por deslocamentos contidos na estrutura MQDH. Se esses deslocamentos resultarem em bytes não utilizados entre uma ou mais da estrutura MQDH, as matrizes de registros e os dados da mensagem, esses bytes não utilizados deverão ser incluídos no valor de *StrucLength*, mas o conteúdo desses bytes não será preservado pelo gerenciador de filas.. É válido para a matriz de registros MQPMR preceder a matriz de registros MQOR.

O valor inicial desse campo é 0.

### ***Codificação (MQLONG) para MQDH***

Esta é a codificação numérica dos dados que seguem as matrizes de registros MQOR e MQPMR; ela não se aplica a dados numéricos na própria estrutura MQDH.

Na chamada MQPUT ou MQPUT1, o aplicativo deve configurar esse campo para o valor apropriado aos dados.

O valor inicial desse campo é 0.

### ***CodedCharSetId (MQLONG) para MQDH***

Este é o identificador do conjunto de caracteres dos dados que seguem as matrizes de registros MQOR e MQPMR; ele não se aplica aos dados de caracteres na própria estrutura MQDH..

Na chamada MQPUT ou MQPUT1, o aplicativo deve configurar esse campo para o valor apropriado aos dados. É possível usar o seguinte valor especial:

#### **MQCCSI\_INHERIT**

Herde o identificador do conjunto de caracteres desta estrutura..

Os dados de caractere nos dados *após* essa estrutura estão no mesmo conjunto de caracteres que essa estrutura

O gerenciador de filas altera esse valor na estrutura enviada na mensagem para o identificador do conjunto de caractere real da estrutura.. Desde que nenhum erro ocorra, a chamada MQGET não retorna o valor MQCCSI\_INHERIT.

Não é possível utilizar MQCCSI\_INHERIT se o valor do campo *PutApplType* em MQMD for MQAT\_BROKER.

Esse valor é suportado nos seguintes ambientes:

-  AIX
-  IBM i
-  Linux
-  Windows

e para clientes IBM MQ conectados a esses sistemas.

O valor inicial desse campo é MQCCSI\_UNDEFINED.

### **Formato (MQCHAR8) para MQDH**

Este é o nome do formato dos dados que seguem as matrizes de registros MQOD e MQPMR (o último que ocorrer).

Na chamada MQPUT ou MQPUT1 , o aplicativo deve configurar esse campo para o valor apropriado aos dados. As regras para codificação desse campo são as mesmas que aquelas para o campo *Format* no MQMD.

O valor inicial desse campo é MQFMT\_NONE.

### **Sinalizadores (MQLONG) para MQDH**

É possível especificar a seguinte sinalização:

#### **MQDHF\_NEW\_MSG\_IDS**

Gere um novo identificador de mensagem para cada destino na lista de distribuição Configure isso somente quando não houver registros de mensagem de colocação presentes ou quando os registros estiverem presentes, mas eles não contiverem o campo *MsgId*

O uso desse sinalizador adia a geração dos identificadores de mensagens até o momento em que a mensagem da lista de distribuição é finalmente dividida em mensagens individuais.. Isso minimiza a quantidade de informações de controle que devem fluir com as mensagens da lista de distribuições.

Quando um aplicativo coloca uma mensagem em uma lista de distribuição, o gerenciador de filas configura MQDHF\_NEW\_MSG\_IDS no MQDH que ele gera quando ambas as instruções a seguir são verdadeiras:

- Não há nenhum registro put-message fornecido pelo aplicativo ou os registros fornecidos não contêm o campo *MsgId*
- O campo *MsgId* em MQMD é MQMI\_NONE ou o campo *Options* em MQPMO inclui MQPMO\_NEW\_MSG\_ID

Se nenhuma sinalização for necessária, especifique o seguinte:

#### **MQDHF\_NONE**

Nenhuma sinalização foi especificada. MQDHF\_NONE é definido para a documentação do programa de auxílio Não se pretende que esta constante seja usada com qualquer outra, mas como seu valor é zero, tal uso não pode ser detectado.

O valor inicial desse campo é MQDHF\_NONE.

### ***PutMsgRecFields (MQLONG) para MQDH***

É possível especificar nenhum ou mais dos seguintes sinalizadores:

#### **MQPMRF\_MSG\_ID**

O campo identificador de mensagem está presente.

#### **MQPMRF\_CORREL\_ID**

O campo do identificador de correlação está presente.

#### **MQPMRF\_GROUP\_ID**

O campo identificador de grupo está presente.

#### **MQPMRF\_FEEDBACK**

Campo de feedback está presente.

#### **MQPMRF\_ACCOUNTING\_TOKEN**

O campo token de contabilidade está presente.

Se nenhum campo MQPMR estiver presente, especifique o seguinte:

#### **MQPMRF\_NONE**

Nenhum campo de registro de mensagem de colocação está presente MQPMRF\_NONE é definido para a documentação do programa de auxílio Não se pretende que esta constante seja usada com qualquer outra, mas como seu valor é zero, tal uso não pode ser detectado.

O valor inicial deste campo é MQPMRF\_NONE.

### ***RecsPresent (MQLONG) para MQDH***

Este é o número de destinos. Uma lista de distribuições deve sempre conter pelo menos um destino, portanto, *RecsPresent* deve sempre ser maior que zero

O valor inicial desse campo é 0.

### ***Deslocamento do ObjectRec(MQLONG) para MQDH***

Isso fornece o deslocamento em bytes do primeiro registro na matriz de registros de objeto MQOR que contém os nomes das filas de destino Há registros *RecsPresent* nesta matriz Esses registros (além de quaisquer bytes ignorados entre o primeiro registro de objeto e o campo anterior) são incluídos no comprimento fornecido pelo campo *StrucLength*.

Uma lista de distribuições deve sempre conter pelo menos um destino, portanto, *ObjectRecOffset* deve sempre ser maior que zero

O valor inicial desse campo é 0.

### ***PutMsgRecOffset (MQLONG) para MQDH***

Isso fornece o deslocamento em bytes do primeiro registro na matriz de registros de mensagens MQPMR put contendo as propriedades de mensagem. Se estiver presente, há registros *RecsPresent* nesta matriz Esses registros (mais quaisquer bytes ignorados entre o primeiro registro de mensagem de entrada e o campo anterior) são incluídos no comprimento fornecido pelo campo *StrucLength*

Registros de mensagens put são opcionais; se nenhum registro for fornecido, *PutMsgRecOffset* será zero e *PutMsgRecFields* terá o valor MQPMRF\_NONE.

O valor inicial desse campo é 0.

## **MQDLH - Cabeçalho de mensagens não entregues**

A estrutura MQDLH descreve as informações que prefixam os dados da mensagem do aplicativo de mensagens na fila de mensagens não entregues. Uma mensagem pode chegar na fila de mensagens não entregues porque o gerenciador de filas ou o agente do canal de mensagem a redirecionou para a fila ou porque um aplicativo colocou a mensagem diretamente na fila..

## Nome do Formato

MQFMT\_DEAD\_LETTER\_HEADER

## Conjunto de caracteres e codificação

Os campos na estrutura MQDLH estão no conjunto de caracteres e na codificação fornecidos pelos campos *CodedCharSetId* e *Encoding*. Eles são especificados na estrutura do cabeçalho que precede o MQDLH ou na estrutura do MQMD se o MQDLH estiver no início dos dados da mensagem do aplicativo.

O conjunto de caracteres deve ser um que tenha caracteres de byte único para os caracteres válidos em nomes de filas.

Se você estiver usando as classes IBM MQ para Java/JMSe a página de códigos definida no MQMD não for suportada pela máquina virtual Java, o MQDLH será gravado no conjunto de caracteres UTF-8.

## Uso

Aplicativos que colocam mensagens diretamente na fila de mensagens não entregues devem prefixar os dados da mensagem com uma estrutura MQDLH e inicializar os campos com valores apropriados. No entanto, o gerenciador de filas não requer que uma estrutura MQDLH esteja presente ou que valores válidos foram especificados para os campos.

Se uma mensagem for muito longa para ser colocada na fila de devoluções, o aplicativo deverá executar um dos seguintes procedimentos:

- Trunque os dados da mensagem para se ajustarem na fila de devoluções
- Registre a mensagem no armazenamento auxiliar e coloque uma mensagem de relatório de exceção na fila de mensagens não entregues indicando isso.
- Descartar a mensagem e retornar um erro para seu originador. Se a mensagem for (ou puder ser) uma mensagem crítica, faça isso somente se for conhecido que o originador ainda possui uma cópia da mensagem; por exemplo, uma mensagem recebida por um agente do canal de mensagens de um canal de comunicação

Qual das ações anteriores é apropriado (se houver) depende do design do aplicativo.

O gerenciador de filas executa processamento especial quando uma mensagem que é um segmento é colocada com uma estrutura MQDLH na frente; consulte a descrição da estrutura MQMDE para obter detalhes adicionais.

## Colocando mensagens na fila de mensagens não entregues

Quando uma mensagem é colocada na fila de mensagens não entregues, a estrutura MQMD usada para a chamada MQPUT ou MQPUT1 deve ser idêntica ao MQMD associado à mensagem (geralmente o MQMD retornado pela chamada MQGET), com exceção do seguinte:

- Configure os campos *CodedCharSetId* e *Encoding* para qualquer conjunto de caractere e codificação usados para campos na estrutura MQDLH.
- Configure o campo *Format* como MQFMT\_DEAD\_LETTER\_HEADER para indicar que os dados iniciam com uma estrutura MQDLH
- Configure os campos de contexto (*AccountingToken*, *ApplIdentityData*, *ApplOriginData*, *PutApplName*, *PutApplType*, *PutDate*, *PutTime*, *UserIdentifier*) usando uma opção de contexto apropriada para as circunstâncias:
  - Um aplicativo que coloca na fila de mensagens não entregues uma mensagem que não está relacionada a nenhuma mensagem anterior deve usar a opção MQPMO\_DEFAULT\_CONTEXT; isso faz com que o gerenciador de filas configure todos os campos de contexto no descritor de mensagem para seus valores padrão

- Um aplicativo servidor que coloca na fila de mensagens não entregues uma mensagem que acabou de receber deve usar a opção MQPMO\_PASS\_ALL\_CONTEXT para preservar as informações de contexto originais.
- Um aplicativo do servidor que coloca na fila de mensagens não entregues uma *resposta* para uma mensagem que ele acabou de receber deve usar a opção MQPMO\_PASS\_IDENTITY\_CONTEXT; isso preserva as informações de identidade, mas configura as informações de origem para serem as do aplicativo do servidor
- Um agente do canal de mensagens que coloca na fila de mensagens não entregues uma mensagem recebida de seu canal de comunicação deve usar a opção MQPMO\_SET\_ALL\_CONTEXT para preservar as informações de contexto originais.

Na própria estrutura MQDLH, configure os campos conforme a seguir:

- Configure os campos *CodedCharSetId*, *Encoding Format* para os valores que descrevem os dados que seguem a estrutura MQDLH, geralmente os valores do descritor de mensagens original
- Configure os campos de contexto *PutApplType*, *PutApplName*, *PutDate* e *PutTime* para valores apropriados ao aplicativo que está colocando a mensagem na fila de devoluções; esses valores não estão relacionados à mensagem original.
- Configure outros campos, conforme apropriado

Certifique-se de que todos os campos tenham valores válidos e que os campos de caracteres sejam preenchidos com espaços em branco para o comprimento definido do campo; não termine os dados de caracteres prematuramente usando um caractere nulo, porque o gerenciador de filas não converte os caracteres nulos e subsequentes em espaços em branco na estrutura MQDLH.

## Obtendo mensagens da fila de mensagens não entregues

Aplicativos que obtêm mensagens da fila de mensagens não entregues devem verificar se as mensagens começam com uma estrutura MQDLH. O aplicativo pode determinar se uma estrutura MQDLH está presente examinando o campo *Format* no descritor de mensagens MQMD; se o campo tiver o valor MQFMT\_DEAD\_LETTER\_HEADER, os dados da mensagem começarão com uma estrutura MQDLH. Esteja ciente também de que as mensagens que os aplicativos obtêm da fila de devoluções podem ser truncadas se originalmente eram muito longas para a fila.

## Campos

**Nota:** Na tabela a seguir, os campos são agrupados por uso e não alfabeticamente. Os tópicos filhos seguem a mesma sequência.

*Tabela 485. Campos em MQDLH para MQDLH.*

Nome e descrição do campo	Nome da constante	Valor inicial (se houver) da constante
<u>StrucId</u> (identificador de estrutura)	MQDLH_STRUC_ID	'DLH↵'
<u>Versão</u> (número da versão da estrutura).	MQDLH_VERSION_1	1
<u>Motivo</u> (mensagem de razão recebida na fila de mensagens não entregues)	MQRC_NONE	0
<u>DestQName</u> (nome da fila de destino original)	Nenhum	Cadeia nula ou espaços em branco
<u>DestQMGrDestino do QMgr</u> (nome do gerenciador de filas de destino original)	Nenhum	Cadeia nula ou espaços em branco
<u>Codificação</u> (codificação numérica de dados que seguem MQDLH)	Nenhum	0

Tabela 485. Campos em MQDLH para MQDLH. (continuação)

Nome e descrição do campo	Nome da constante	Valor inicial (se houver) da constante
<u>CodedCharSetId</u> (identificador do conjunto de caracteres de dados que segue MQDLH)..	MQCCSI_UNDEFINED	0
<u>Formato</u> (nome do formato de dados após MQDLH)	MQFMT_NONE	Espaços em branco
<u>PutApplTipo</u> (tipo de aplicativo que coloca a mensagem na fila de mensagens não entregues)	Nenhum	0
<u>PutApplNome</u> (nome do aplicativo que coloca a mensagem na fila de devoluções).	Nenhum	Cadeia nula ou espaços em branco
<u>PutDate</u> (data em que a mensagem foi colocada na fila de inatividade)	Nenhum	Cadeia nula ou espaços em branco
<u>PutTime</u> (horário em que a mensagem foi colocada na fila de devoluções)..	Nenhum	Cadeia nula ou espaços em branco

**Notas:**

1. O símbolo ~ representa um único caractere em branco.
2. O valor Sequência nula ou espaços em branco indica a sequência nula em C e caracteres em branco em outras linguagens de programação.
3. Na linguagem de programação C, a variável macroMQDLH\_DEFAULT contém os valores que são listados na tabela. Use-o da seguinte maneira para fornecer valores iniciais para os campos na estrutura:

```
MQDLH MyDLH = {MQDLH_DEFAULT};
```

## Declarações de idiomas

Declaração C para MQDLH

```
typedef struct tagMQDLH MQDLH;
struct tagMQDLH {
    MQCHAR4   StrucId;           /* Structure identifier */
    MQLONG    Version;          /* Structure version number */
    MQLONG    Reason;           /* Reason message arrived on dead-letter
                               (undelivered-message) queue */
    MQCHAR48  DestQName;        /* Name of original destination queue */
    MQCHAR48  DestQMgrName;     /* Name of original destination queue
                               manager */
    MQLONG    Encoding;         /* Numeric encoding of data that follows
                               MQDLH */
    MQLONG    CodedCharSetId;   /* Character set identifier of data that
                               follows MQDLH */
    MQCHAR8   Format;           /* Format name of data that follows
                               MQDLH */
    MQLONG    PutApplType;      /* Type of application that put message on
                               dead-letter (undelivered-message)
                               queue */
    MQCHAR28  PutApplName;      /* Name of application that put message on
                               dead-letter (undelivered-message)
                               queue */
    MQCHAR8   PutDate;          /* Date when message was put on dead-letter
                               (undelivered-message) queue */
    MQCHAR8   PutTime;          /* Time when message was put on the
                               dead-letter (undelivered-message)
                               queue */
};
```



## Declaração COBOL para MQDLH

```
** MQDLH structure
10 MQDLH.
** Structure identifier
15 MQDLH-STRUCID PIC X(4).
** Structure version number
15 MQDLH-VERSION PIC S9(9) BINARY.
** Reason message arrived on dead-letter (undelivered-message) queue
15 MQDLH-REASON PIC S9(9) BINARY.
** Name of original destination queue
15 MQDLH-DESTQNAME PIC X(48).
** Name of original destination queue manager
15 MQDLH-DESTQMGRNAME PIC X(48).
** Numeric encoding of data that follows MQDLH
15 MQDLH-ENCODING PIC S9(9) BINARY.
** Character set identifier of data that follows MQDLH
15 MQDLH-CODEDCHARSETID PIC S9(9) BINARY.
** Format name of data that follows MQDLH
15 MQDLH-FORMAT PIC X(8).
** Type of application that put message on dead-letter
** (undelivered-message) queue
15 MQDLH-PUTAPPLTYPE PIC S9(9) BINARY.
** Name of application that put message on dead-letter
** (undelivered-message) queue
15 MQDLH-PUTAPPLNAME PIC X(28).
** Date when message was put on dead-letter (undelivered-message)
** queue
15 MQDLH-PUTDATE PIC X(8).
** Time when message was put on the dead-letter (undelivered-message)
** queue
15 MQDLH-PUTTIME PIC X(8).
```

## Declaração PL/I para MQDLH

```
dcl
1 MQDLH based,
3 StrucId char(4), /* Structure identifier */
3 Version fixed bin(31), /* Structure version number */
3 Reason fixed bin(31), /* Reason message arrived on
dead-letter (undelivered-message)
queue */
3 DestQName char(48), /* Name of original destination
queue */
3 DestQMgrName char(48), /* Name of original destination queue
manager */
3 Encoding fixed bin(31), /* Numeric encoding of data that
follows MQDLH */
3 CodedCharSetId fixed bin(31), /* Character set identifier of data
that follows MQDLH */
3 Format char(8), /* Format name of data that follows
MQDLH */
3 PutApplType fixed bin(31), /* Type of application that put
message on dead-letter
(undelivered-message) queue */
3 PutApplName char(28), /* Name of application that put
message on dead-letter
(undelivered-message) queue */
3 PutDate char(8), /* Date when message was put on
dead-letter (undelivered-message)
queue */
3 PutTime char(8); /* Time when message was put on the
dead-letter (undelivered-message)
queue */
```

## Declaração High Level Assembler para MQDLH

MQDLH	DSECT	
MQDLH_STRUCID	DS CL4	Structure identifier
MQDLH_VERSION	DS F	Structure version number
MQDLH_REASON	DS F	Reason message arrived on dead-letter
*		(undelivered-message) queue
MQDLH_DESTQNAME	DS CL48	Name of original destination queue
MQDLH_DESTQMGRNAME	DS CL48	Name of original destination queue
*		manager

MQDLH_ENCODING	DS	F	Numeric encoding of data that follows MQDLH
* MQDLH_CODEDCHARSETID	DS	F	Character set identifier of data that follows MQDLH
* MQDLH_FORMAT	DS	CL8	Format name of data that follows MQDLH
* MQDLH_PUTAPPLTYPE	DS	F	Type of application that put message on dead-letter (undelivered-message) queue
* MQDLH_PUTAPPLNAME	DS	CL28	Name of application that put message on dead-letter (undelivered-message) queue
* MQDLH_PUTDATE	DS	CL8	Date when message was put on dead-letter (undelivered-message) queue
* MQDLH_PUTTIME	DS	CL8	Time when message was put on the dead-letter (undelivered-message) queue
* MQDLH_LENGTH	EQU	*-MQDLH	
	ORG	MQDLH	
* MQDLH_AREA	DS	CL(MQDLH_LENGTH)	

## Declaração Visual Basic para MQDLH

```

Type MQDLH
  StrucId      As String*4  'Structure identifier'
  Version     As Long      'Structure version number'
  Reason      As Long      'Reason message arrived on dead-letter'
  Reason      As Long      '(undelivered-message) queue'
  DestQName   As String*48 'Name of original destination queue'
  DestQMgrName As String*48 'Name of original destination queue'
  DestQMgrName As String*48 'manager'
  Encoding    As Long      'Numeric encoding of data that follows'
  Encoding    As Long      'MQDLH'
  CodedCharSetId As Long    'Character set identifier of data that'
  CodedCharSetId As Long    'follows MQDLH'
  Format      As String*8  'Format name of data that follows MQDLH'
  PutApplType As Long      'Type of application that put message on'
  PutApplType As Long      'dead-letter (undelivered-message) queue'
  PutApplName As String*28 'Name of application that put message on'
  PutApplName As String*28 'dead-letter (undelivered-message) queue'
  PutDate     As String*8  'Date when message was put on dead-letter'
  PutDate     As String*8  '(undelivered-message) queue'
  PutTime     As String*8  'Time when message was put on the'
  PutTime     As String*8  'dead-letter (undelivered-message) queue'
End Type

```

### **StrucId (MQCHAR4) para MQDLH**

Este é o identificador da estrutura do cabeçalho de devoluções. Ele é sempre um campo de entrada. Seu valor é MQDLH\_STRUC\_ID.

O valor deve ser:.

#### **MQDLH\_STRUC\_ID**

Identificador para a estrutura do cabeçalho de devoluções.

Para a linguagem de programação C, a constante MQDLH\_STRUC\_ID\_ARRAY também é definida. Ele tem o mesmo valor que MQDLH\_STRUC\_ID, mas é uma matriz de caracteres em vez de uma cadeia.

### **Versão (MQLONG) para MQDLH**

A versão é o número da versão da estrutura

O valor deve ser:.

#### **MQDLH\_VERSION\_1**

Número da versão para a estrutura do cabeçalho de devoluções

A constante a seguir especifica o número da versão atual:

#### **MQDLH\_CURRENT\_VERSION**

A versão atual da estrutura do cabeçalho de devoluções

O valor inicial desse campo é MQDLH\_VERSION\_1.

### **Motivo (MQLONG) para MQDLH**

O campo Razão identifica o motivo pelo qual a mensagem foi colocada na fila de mensagens não entregues em vez de na fila de destino original.

Isso identifica o motivo pelo qual a mensagem foi colocada na fila de devoluções em vez de na fila de destino original. Deve ser um dos valores MQFB\_\* ou MQRC\_\* (por exemplo, MQRC\_Q\_FULL). Consulte a descrição do campo *Feedback* em “MQMD - descritor de mensagem” na página 435 para obter detalhes sobre os valores MQFB\_\* comuns que podem ocorrer

Se o valor estiver no intervalo de MQFB\_IMS\_FIRST a MQFB\_IMS\_LAST, o código de erro real IMS poderá ser determinado subtraindo MQFB\_IMS\_ERROR do valor do campo *Reason*.

Alguns valores MQFB\_\* ocorrem apenas neste campo. Eles se relacionam a mensagens do repositório, mensagens acionadoras ou mensagens da fila de transmissão que foram transferidas para a fila de mensagens não entregues São elas:

#### **MQFB\_APPL\_CANNOT\_BE\_STARTED ( X'00000109' )**

Um aplicativo processando uma mensagem do acionador não pode iniciar o aplicativo nomeado no campo *AppId* da mensagem do acionador (consulte “MQTM-Mensagem do acionador” na página 622).

No z/OS, a transação CKTI CICS é um exemplo de um aplicativo que processa mensagens acionadoras

#### **MQFB\_APPL\_TYPE\_ERROR ( X'0000010B' )**

Um aplicativo que processa uma mensagem do acionador não pode iniciar o aplicativo porque o campo *AppType* da mensagem do acionador não é válido (consulte “MQTM-Mensagem do acionador” na página 622).

No z/OS, a transação CKTI CICS é um exemplo de um aplicativo que processa mensagens acionadoras

#### **MQFB\_BIND\_OPEN\_CLUSRCVR\_DEL ( X'00000119' )**

A mensagem estava no sistema SYSTEM.CLUSTER.TRANSMIT.QUEUE destinado a uma fila de clusters que foi aberta com a opção MQOO\_BIND\_ON\_OPEN, mas o canal do receptor de clusters remoto a ser usado para transmitir a mensagem para a fila de destino foi excluído antes que a mensagem pudesse ser enviada Como MQOO\_BIND\_ON\_OPEN foi especificado, apenas o canal selecionado quando a fila foi aberta pode ser usado para transmitir a mensagem.. Como esse canal não está mais disponível, a mensagem é colocada em uma fila de mensagens não entregues..

#### **MQFB\_NOT\_A\_REPOSITORY\_MSG ( X'00000118' )**

A mensagem não é uma mensagem de repositórios

#### **MQFB\_STOPPED\_BY\_CHAD\_EXIT ( X'00000115' )**

A mensagem foi interrompida pela saída de definição automática de canal.

#### **MQFB\_STOPPED\_BY\_MSG\_EXIT ( X'0000010D' )**

A mensagem foi interrompida pela saída de mensagem do canal..

#### **MQFB\_TM\_ERROR ( X'0000010A' )**

O campo *Format* em MQMD especifica MQFMT\_TRIGGER, mas a mensagem não começa com uma estrutura MQTM válida Por exemplo, o destaque mnemônico *StrucId* pode não ser válido, o *Version* pode não ser reconhecido ou o comprimento da mensagem do acionador pode ser insuficiente para conter a estrutura MQTM.

No z/OS, a transação CKTI CICS é um exemplo de um aplicativo que processa mensagens acionadoras e pode gerar esse código de feedback

#### **MQFB\_XMIT\_Q\_MSG\_ERROR ( X'0000010F' )**

Um agente do canal de mensagens descobriu que uma mensagem na fila de transmissão não está no formato correto O agente do canal de mensagens coloca a mensagem na fila de mensagens não entregues usando esse código de feedback...

Uma causa comum é que uma mensagem foi colocada diretamente na fila de transmissão, portanto, a mensagem não tem o cabeçalho XQH esperado. Mensagens devem ser colocadas em uma fila de transmissão por meio de uma fila remota, a menos que o aplicativo construa o cabeçalho MQXQH.

O valor inicial desse campo é MQRC\_NONE.

### **DestQName (MQCHAR48) para MQDLH**

DestQName é o nome da fila de mensagens que foi o destino original da mensagem.

O comprimento desse campo é fornecido por MQ\_Q\_NAME\_LENGTH O valor inicial deste campo é a cadeia nula em C e 48 caracteres em branco em outras linguagens de programação.

### **DestQMgrNome (MQCHAR48) para MQDLH**

DestQMgrNome é o nome do gerenciador de filas que era o destino original da mensagem.

O comprimento desse campo é fornecido por MQ\_Q\_MGR\_NAME\_LENGTH O valor inicial deste campo é a cadeia nula em C e 48 caracteres em branco em outras linguagens de programação.

### **Codificação (MQLONG) para MQDLH**

A codificação é a codificação numérica dos dados que seguem a estrutura MQDLH (geralmente os dados da mensagem original); ela não se aplica a dados numéricos na própria estrutura MQDLH.

Na chamada MQPUT ou MQPUT1 , o aplicativo deve configurar esse campo para o valor apropriado aos dados.

O valor inicial desse campo é 0.

### **CodedCharSetId (MQLONG) para MQDLH**

CodedCharSetId é o identificador do conjunto de caracteres dos dados que fluem pela estrutura MQDLH (geralmente os dados da mensagem original); ele não se aplica aos dados de caracteres na própria estrutura MQDLH.

Na chamada MQPUT ou MQPUT1 , o aplicativo deve configurar esse campo para o valor apropriado aos dados. O seguinte valor especial pode ser usado:

#### **MQCCSI\_INHERIT**

Os dados de caracteres nos dados que seguem esta estrutura estão no mesmo conjunto de caracteres que esta estrutura

O gerenciador de filas altera esse valor na estrutura enviada na mensagem para o identificador do conjunto de caractere real da estrutura.. Desde que não ocorra nenhum erro, o valor MQCCSI\_INHERIT não será retornado pela chamada MQGET

Não é possível utilizar MQCCSI\_INHERIT se o valor do campo *PutApplType* em MQMD for MQAT\_BROKER.

Esse valor é suportado nos seguintes ambientes:

-  AIX
-  IBM i
-  Linux
-  Windows

e para clientes IBM MQ conectados a esses sistemas.

O valor inicial desse campo é MQCCSI\_UNDEFINED.

### **Formato (MQCHAR8) para MQDLH**

Format é o nome do formato dos dados que seguem a estrutura MQDLH (geralmente os dados da mensagem original).

Na chamada MQPUT ou MQPUT1 , o aplicativo deve configurar esse campo para o valor apropriado aos dados. As regras para codificar esse campo são as mesmas que aquelas regras para codificar o campo *Format* no MQMD

O comprimento desse campo é fornecido por MQ\_FORMAT\_LENGTH O valor inicial desse campo é MQFMT\_NONE.

### **PutApplTipo (MQLONG) para MQDLH**

PutApplTipo é o tipo de aplicativo que coloca a mensagem na fila de mensagens não entregues (não entregues).

Esse campo tem o mesmo significado que o campo *PutApplType* no descritor de mensagens MQMD (consulte “MQMD - descritor de mensagem” na página 435 para obter detalhes).

Se o gerenciador de filas redirecionar a mensagem para a fila de mensagens não entregues, *PutApplType* terá o valor MQAT\_QMGR.

O valor inicial desse campo é 0.

### **PutApplNome (MQCHAR28) para MQDLH**

PutApplNome é o nome do aplicativo que colocou a mensagem na fila de mensagens não entregues (não entregues).

O formato do nome depende do campo *PutApplType* .. O formato pode variar liberação para liberação. Consulte a descrição do campo *PutApplName* em “MQMD - descritor de mensagem” na página 435

Se o gerenciador de filas redirecionar a mensagem para a fila de mensagens não entregues, o *PutApplName* conterá os primeiros 28 caracteres do nome do gerenciador de filas, preenchidos com espaços em branco, se necessário.

O comprimento desse campo é fornecido por MQ\_PUT\_APPL\_NAME\_LENGTH O valor inicial deste campo é a sequência nula em C e 28 caracteres em branco em outras linguagens de programação.

### **PutDate (MQCHAR8) para MQDLH.**

PutDate é a data em que a mensagem foi colocada na fila de mensagens não entregues.

O formato utilizado para a data em que esse campo é gerado pelo gerenciador de filas é:

- AAAAMMDD

em que os caracteres representam:

#### **AAAA**

ano (quatro dígitos numéricos)

#### **MM**

mês do ano (01 a 12)

#### **DD**

dia do mês (01 a 31)

O Tempo Médio de Greenwich (GMT) é usado para os campos *PutDate* e *PutTime* , sujeito ao relógio do sistema sendo configurado com precisão para GMT.

O comprimento desse campo é fornecido por MQ\_PUT\_DATE\_LENGTH O valor inicial desse campo é a sequência nula em C e oito caracteres em branco em outras linguagens de programação.

### **PutTime (MQCHAR8) para MQDLH**

PutTime é o horário em que a mensagem foi colocada na fila de mensagens não entregues.

O formato usado para o horário em que esse campo é gerado pelo gerenciador de filas é:

- HHMMSSTH

em que os caracteres representam:

**DOMS**

horas (00 a 23)

**MM**

minutos (00 a 59)

**SS**

segundos (00 a 59; ver nota)

**T**

Décimos de segundo (0 a 9)

**H**

centésimos de segundo (0 a 9)

**Nota:** Se o relógio do sistema for sincronizado com um padrão de tempo muito preciso, será possível em raras ocasiões que 60 ou 61 sejam retornados para os segundos em *PutTime*. Isso acontece quando os segundos bissextos são inseridos no padrão de tempo global.

O Tempo Médio de Greenwich (GMT) é usado para os campos *PutDate* e *PutTime*, sujeito ao relógio do sistema sendo configurado com precisão para GMT.

O comprimento desse campo é fornecido por MQ\_PUT\_TIME\_LENGTH. O valor inicial desse campo é a sequência nula em C e oito caracteres em branco em outras linguagens de programação.

## MQDMHO-Excluir opções do identificador de mensagens

A estrutura **MQDMHO** permite que os aplicativos especifiquem opções que controlam como as manipulações de mensagens são excluídas. A estrutura é um parâmetro de entrada na chamada de **MQDLTMH**.

### Conjunto de caracteres e codificação

Os dados em **MQDMHO** devem estar no conjunto de caracteres do aplicativo e codificação do aplicativo (**MQENC\_NATIVE**).

### Campos

**Nota:** Na tabela a seguir, os campos são agrupados por uso e não alfabeticamente. Os tópicos filhos seguem a mesma sequência.

Nome e descrição do campo	Nome da constante	Valor inicial (se houver) da constante
<u>StrucId</u> (identificador de estrutura)	MQDMHO_STRUC_ID	'DMHO'
<u>Versão</u> (número da versão da estrutura).	MQDMHO_VERSION_1	1
<u>Opções</u> (opções).	MQDMHO_NONE	0
<b>Notas:</b> 1. Na linguagem de programação C, a variável macro <code>MQDMHO_DEFAULT</code> contém os valores que são listados na tabela. Ele pode ser usado da seguinte maneira para fornecer valores iniciais para os campos na estrutura: <pre>MQDMHO MyDMHO = {MQDMHO_DEFAULT};</pre>		

## Declarações de idiomas

### Declaração C para MQDMHO

```
typedef struct tagMQDMHO;
struct tagMQDMHO {
    MQCHAR4   StrucId;           /* Structure identifier */
    MQLONG    Version;          /* Structure version number */
    MQLONG    Options;          /* Options that control the action of MQDLTMH */
};
```

### Declaração COBOL para MQDMHO

```
** MQDMHO structure
10 MQDMHO.
** Structure identifier
15 MQDMHO-STRUCID PIC X(4).
** Structure version number
15 MQDMHO-VERSION PIC S9(9) BINARY.
** Options that control the action of MQDLTMH
15 MQDMHO-OPTIONS PIC S9(9) BINARY.
```

### Declaração PL/I para MQDMHO

```
dcl
1 MQDMHO based,
3 StrucId char(4), /* Structure identifier */
3 Version fixed bin(31), /* Structure version number */
3 Options fixed bin(31), /* Options that control the action of MQDLTMH */
```

### Declaração High Level Assembler para MQDMHO

```
MQDMHO          DSECT
MQDMHO_STRUCID  DS CL4 Structure identifier
MQDMHO_VERSION  DS F   Structure version number
MQDMHO_OPTIONS  DS F   Options that control the action of
*               MQDLTMH
MQDMHO_LENGTH   EQU *-MQDMHO
MQDMHO_AREA     DS CL(MQDMHO_LENGTH)
```

### **StrucId (MQCHAR4) para MQDMHO**

Este é o identificador de estrutura da estrutura de opções de manipulação de mensagens de exclusão... Ele é sempre um campo de entrada Seu valor é MQDMHO\_STRUC\_ID.

O valor deve ser:

#### **MQDMHO\_STRUC\_ID**

Identificador para a estrutura de opções de manipulação de mensagem de exclusão.

Para a linguagem de programação C, a constante **MQDMHO\_STRUC\_ID\_ARRAY** também é definida.. Isso tem o mesmo valor que **MQDMHO\_STRUC\_ID**, mas é uma matriz de caracteres em vez de uma sequência..

### **Versão (MQLONG) para MQDMHO**

Este é o número da versão da estrutura; o valor deve ser:

#### **MQDMHO\_VERSION\_1**

Version-1 exclua a estrutura de opções de manipulação de mensagens

A constante a seguir especifica o número da versão atual:

#### **MQDMHO\_CURRENT\_VERSION**

A versão atual da estrutura de opções de manipulação de mensagens de exclusão

Isto é sempre um campo de entrada. O valor inicial desse campo é **MQDMHO\_VERSION\_1**.

## Opções (MQLONG) para MQDMHO

O valor deve ser:.

### **MQDMHO\_NONE**

Nenhuma opção especificada.

Isto é sempre um campo de entrada. O valor inicial desse campo é **MQDMHO\_NONE**.

## MQDMPO-Excluir opções de propriedade de mensagens

A estrutura MQDMPO permite que aplicativos especifiquem opções que controlam como as propriedades de mensagens são excluídas. A estrutura é um parâmetro de entrada na chamada de MQDLTMP

## Conjunto de caracteres e codificação

Os dados no MQDMPO devem estar no conjunto de caractere do aplicativo e na codificação do aplicativo (MQENC\_NATIVE)..

## Campos

**Nota:** Na tabela a seguir, os campos são agrupados por uso e não alfabeticamente. Os tópicos filhos seguem a mesma sequência.

Tabela 487. Campos em MQDMPO		
Nome e descrição do campo	Nome da constante	Valor inicial (se houver) da constante
StrucId (identificador de estrutura)	MQDMPO_STRUC_ID	'DMPO'
Versão (número da versão da estrutura).	MQDMPO_VERSION_1	1
Opções (opções que controlam a ação de MQDMPO)	Opções que controlam a ação de MQDLTMP	MQDMPO_NONE

**Notas:**

1. Na linguagem de programação C, a variável macroMQDMPO\_DEFAULT contém os valores listados na tabela. Use-o da seguinte maneira para fornecer valores iniciais para os campos na estrutura:

```
MQDMPO MyDMPO = {MQDMPO_DEFAULT};
```

## Declarações de idiomas

Declaração C para MQDMPO

```
typedef struct tagMQDMPO MQDMPO;
struct tagMQDMPO {
    MQCHAR4  StrucId;          /* Structure identifier */
    MQLONG   Version;         /* Structure version number */
    MQLONG   Options;         /* Options that control the action of
                             MQDLTMP */
};
```

Declaração COBOL para MQDMPO

```
** MQDMPO structure
```



```

10 MQDMPO.
**  Structure identifier
   15 MQDMPO-STRUCID          PIC X(4).
**  Structure version number
   15 MQDMPO-VERSION         PIC S9(9) BINARY.
**  Options that control the action of MQDLTMP
   15 MQDMPO-OPTIONS        PIC S9(9) BINARY.

```

### Declaração PL/I para MQDMPO

```

Dcl
  1 MQDMPO based,
  3 StrucId      char(4),          /* Structure identifier */
  3 Version      fixed bin(31),   /* Structure version number */
  3 Options      fixed bin(31),   /* Options that control the action
                                of MQDLTMP */

```

### Declaração do High Level Assembler para MQDMPO

```

MQDMPO          DSECT
MQDMPO_STRUCID  DS   CL4  Structure identifier
MQDMPO_VERSION  DS   F    Structure version number
MQDMPO_OPTIONS  DS   F    Options that control the
*                action of MQDLTMP
MQDMPO_LENGTH   EQU   *-MQDMPO
MQDMPO_AREA     DS   CL(MQDMPO_LENGTH)

```

### **StrucId (MQCHAR4) para MQDMPO**

Este é o identificador de estrutura da estrutura de opções de propriedades de mensagem de exclusão... Ele é sempre um campo de entrada Seu valor é MQDMPO\_STRUC\_ID.

O valor deve ser:.

#### **MQDMPO\_STRUC\_ID**

Identificador para a estrutura de opções excluir propriedade de mensagem.

Para a linguagem de programação C, a constante MQDMPO\_STRUC\_ID\_ARRAY também é definida. Este tem o mesmo valor que MQDMPO\_STRUC\_ID, mas é uma matriz de caracteres em vez de uma sequência.

### **Versão (MQLONG) para MQDMPO**

Excluir a estrutura de opções da propriedade de mensagem-campo Versão

Este é o número da versão da estrutura O valor deve ser:.

#### **MQDMPO\_VERSION\_1**

Número da versão para a estrutura de opções de propriedades de mensagens de exclusão

A constante a seguir especifica o número da versão atual:

#### **MQDMPO\_CURRENT\_VERSION**

Versão atual da estrutura de opções de propriedades de mensagem de exclusão

Isto é sempre um campo de entrada. O valor inicial desse campo é MQDMPO\_VERSION\_1.

### **Opções (MQLONG) para MQDMPO**

Excluir estrutura de opções de propriedade de mensagens-campo Opções

**Opções de local:** As opções a seguir estão relacionadas ao local relativo da propriedade comparado ao cursor da propriedade.

#### **MQDMPO\_DEL\_FIRST**

Exclui a primeira propriedade que corresponde ao nome especificado

## **MQDMPO\_DEL\_PROP\_UNDER\_CURSOR**

Exclui a propriedade apontada pelo cursor da propriedade; essa é a última propriedade consultada usando a opção MQIMPO\_INQ\_FIRST ou MQIMPO\_INQ\_NEXT.

O cursor de propriedade é reconfigurado quando o identificador de mensagem é reutilizado. Ele também é reconfigurado quando a manipulação de mensagens é especificada no campo *MsgHandle* da estrutura MQGMO em uma chamada MQGET, ou na estrutura MQPMO em uma chamada MQPUT.

Se essa opção for usada quando o cursor da propriedade ainda não tiver sido estabelecido, a chamada falhará com o código de conclusão MQCC\_FAILED, e a razão MQRC\_PROPERTY\_NOT\_AVAILABLE. Se a propriedade apontada pelo cursor da propriedade já tiver sido excluída, a chamada também falhará com o código de conclusão MQCC\_FAILED e a razão MQRC\_PROPERTY\_NOT\_AVAILABLE.

Se nenhuma das opções for necessária, a opção a seguir poderá ser usada:

## **MQDMPO\_NONE**

Nenhuma opção especificada.

Esse campo é sempre um campo de entrada. O valor inicial desse campo é MQDMPO\_DEL\_FIRST.

## **MQEPH - cabeçalho PCF integrado**

A estrutura MQEPH descreve os dados adicionais presentes em uma mensagem quando essa mensagem é uma mensagem de formato de comando programável (PCF). O campo *PCFHeader* define os parâmetros PCF que seguem essa estrutura e isso permite seguir os dados da mensagem PCF com outros cabeçalhos.

### **Nome do Formato**

MQFMT\_EMBEDDED\_PCF

### **Conjunto de caracteres e codificação**

Os dados em MQEPH devem estar no conjunto de caracteres fornecido pelo atributo do gerenciador de filas do **CodedCharSetId** e pela codificação do gerenciador de fila local fornecido por MQENC\_NATIVE.

Configure o conjunto de caracteres e a codificação do MQEPH nos campos *CodedCharSetId* e *Encoding* no MQMD (se a estrutura MQEPH estiver no início dos dados da mensagem) ou na estrutura do cabeçalho que precede a estrutura MQEPH (todos os outros casos).

### **Uso**

Não é possível usar estruturas MQEPH para enviar comandos para o servidor de comando ou qualquer outro servidor de aceitação PCF do gerenciador de filas.

Da mesma forma, o servidor de comandos ou qualquer outro servidor de aceitação PCF do gerenciador de filas não gera respostas ou eventos contendo estruturas MQEPH.

### **Campos**

**Nota:** Na tabela a seguir, os campos são agrupados por uso e não alfabeticamente. Os tópicos filhos seguem a mesma sequência.

<i>Tabela 488. Campos em MQEPH para MQEPH</i>		
<b>Nome e descrição do campo</b>	<b>Nome da constante</b>	<b>Valor inicial (se houver) da constante</b>
<u>StrucId</u> (identificador de estrutura)	MQEPH_STRUC_ID	' EPH↵ '
<u>Versão</u> (número da versão da estrutura).	MQEPH_VERSION_1	1

Tabela 488. Campos em MQEPH para MQEPH (continuação)

Nome e descrição do campo	Nome da constante	Valor inicial (se houver) da constante
<u>StrucLength</u> (comprimento da estrutura MQEPH mais as estruturas MQCFH e de parâmetro que a seguem)	MQEPH_STRUC_LEN H_FIXED	68
Codificação (codificação numérica de dados que seguem a última estrutura de parâmetro PCF).	Nenhum	0
<u>CodedCharSetId</u> (identificador do conjunto de caracteres de dados que segue a última estrutura de parâmetro PCF)..	MQCCSI_UNDEFINED	0
<u>Formato</u> (nome do formato de dados que segue a última estrutura de parâmetro PCF)..	MQFMT_NONE	Espaços em branco
<u>Sinalizadores</u> (sinalizadores)	MQEPH_NONE	0
<u>PCFHeader</u> (cabeçalho de formato de comando programável (PCF))	Nomes e valores conforme definido em <a href="#">Tabela 489 na página 378</a>	0

**Notas:**

1. O símbolo ~ representa um único caractere em branco.
2. Na linguagem de programação C, a variável macro MQEPH\_DEFAULT contém os valores listados na tabela. Use-o da seguinte maneira para fornecer valores iniciais para os campos na estrutura:

```
MQEPH MyEPH = {MQEPH_DEFAULT};
```

## Declarações de idiomas

### Declaração C para MQEPH

```
typedef struct tagMQEPH MQEPH;
struct tagMQDH {
    MQCHAR4  StrucId;           /* Structure identifier */
    MQLONG   Version;          /* Structure version number */
    MQLONG   StrucLength;      /* Total length of MQEPH including the MQCFH
                               and parameter structures that follow it */
    MQLONG   Encoding;         /* Numeric encoding of data that follows last
                               PCF parameter structure */
    MQLONG   CodedCharSetId;   /* Character set identifier of data that
                               follows last PCF parameter structure */
    MQCHAR8  Format;           /* Format name of data that follows last PCF
                               parameter structure */
    MQLONG   Flags;            /* Flags */
    MQCFH    PCFHeader;        /* Programmable command format header */
};
```

### Declaração COBOL para MQEPH

```
** MQEPH structure
10 MQEPH.
** Structure identifier
15 MQEPH-STRUCID PIC X(4).
** Structure version number
15 MQEPH-VERSION PIC S9(9) BINARY.
** Total length of MQEPH structure including the MQCFH
** and parameter structures that follow it
15 MQEPH-STRUCLNGTH PIC S9(9) BINARY.
```

```

**      Numeric encoding of data that follows last
**      PCF structure
15 MQEPH-ENCODING          PIC S9(9) BINARY.
**      Character set identifier of data that
**      follows last PCF parameter structure
15 MQEPH-CODEDCHARSETID  PIC S9(9) BINARY.
**      Format name of data that follows last PCF
**      parameter structure
15 MQEPH-FORMAT          PIC X(8).
**      Flags
15 MQEPH-FLAGS           PIC S9(9) BINARY.
**      Programmable command format header
15 MQEPH-PCFHEADER.
**      Structure type
20 MQEPH-PCFHEADER-TYPE  PIC S9(9) BINARY.
**      Structure length
20 MQEPH-PCFHEADER-STRULENGTH  PIC S9(9) BINARY.
**      Structure version number
20 MQEPH-PCFHEADER-VERSION  PIC S9(9) BINARY.
**      Command identifier
20 MQEPH-PCFHEADER-COMMAND  PIC S9(9) BINARY.
**      Message sequence number
20 MQEPH-PCFHEADER-MSGSEQNUMBER  PIC S9(9) BINARY.
**      Control options
20 MQEPH-PCFHEADER-CONTROL  PIC S9(9) BINARY.
**      Completion code
20 MQEPH-PCFHEADER-COMPCODE  PIC S9(9) BINARY.
**      Reason code qualifying completion code
20 MQEPH-PCFHEADER-REASON  PIC S9(9) BINARY.
**      Count of parameter structures
20 MQEPH-PCFHEADER-PARAMETERCOUNT  PIC S9(9) BINARY.

```

## Declaração PL/I para MQEPH

```

dcl
  1 MQEPH based,
  3 StrucId          char(4),          /* Structure identifier */
  3 Version          fixed bin(31),    /* Structure version number */
  3 StrucLength      fixed bin(31),    /* Total length of MQEPH including the
                                     MQCFH and parameter structures that
                                     follow it
  3 Encoding         fixed bin(31),    /* Numeric encoding of data that follows
                                     last PCF parameter structure
  3 CodedCharSetId  fixed bin(31),    /* Character set identifier of data that
                                     follows last PCF parameter structure
  3 Format           char(8),          /* Format name of data that follows last
                                     PCF parameter structure */
  3 Flags           fixed bin(31),    /* Flags */
  3 PCFHeader,      /* Programmable command format header
  5 Type           fixed bin(31),    /* Structure type */
  5 StrucLength     fixed bin(31),    /* Structure length */
  5 Version         fixed bin(31),    /* Structure version number */
  5 Command         fixed bin(31),    /* Command identifier */
  5 MsgseqNumber    fixed bin(31),    /* Message sequence number */
  5 Control         fixed bin(31),    /* Control options */
  5 CompCode        fixed bin(31),    /* Completion code */
  5 Reason          fixed bin(31),    /* Reason code qualifying completion code */
  5 ParameterCount  fixed bin(31);    /* Count of parameter structures */

```

## Declaração High Level Assembler para MQEPH

MQEPH	DSECT		
MQEPH_STRUCID	DS	CL4	Structure identifier
MQEPH_VERSION	DS	F	Structure version number
MQEPH_STRULENGTH	DS	F	Total length of MQEPH including the
*			MQCFH and parameter structures that
			follow it
MQEPH_ENCODING	DS	F	Numeric encoding of data that follows
*			last PCF parameter structure
MQEPH_CODEDCHARSETID	DS	F	Character set identifier of data that
*			follows last PCF parameter structure
MQEPH_FORMAT	DS	CL8	Format name of data that follows last
*			PCF parameter structure
MQEPH_FLAGS	DS	F	Flags
MQEPH_PCFHEADER	DS	OF	Force fullword alignment
MQEPH_PCFHEADER_TYPE	DS	F	Structure type
MQEPH_PCFHEADER_STRULENGTH	DS	F	Structure length

```

MQEPH_PCFHEADER_VERSION      DS   F   Structure version number
MQEPH_PCFHEADER_COMMAND     DS   F   Command identifier
MQEPH_PCFHEADER_MSGSEQNUMBER DS   F   Structure length
MQEPH_PCFHEADER_CONTROL     DS   F   Control options
MQEPH_PCFHEADER_COMPCODE    DS   F   Completion code
MQEPH_PCFHEADER_REASON      DS   F   Reason code qualifying completion code
MQEPH_PCFHEADER_PARAMETER COUNT DS   F   Count of parameter structures
MQEPH_PCFHEADER_LENGTH      EQU  *-MQEPH_PCFHEADER
                              ORG  MQEPH_PCFHEADER
MQEPH_PCFHEADER_AREA        DS   CL(MQEPH_PCFHEADER_LENGTH)
*
MQEPH_LENGTH                EQU  *-MQEPH
                              ORG  MQEPH
MQEPH_AREA                   DS   CL(MQEPH_LENGTH)

```

## Declaração Visual Basic para MQEPH

```

Type MQEPH
  StrucId      As String*4 'Structure identifier'
  Version     As Long      'Structure version number'
  StrucLength  As Long      'Total length of MQEPH structure including the MQCFH'
                              'and parameter structures that follow it'
  Encoding     As Long      'Numeric encoding of data that follows last'
                              'PCF parameter structure'
  CodedCharSetId As Long    'Character set identifier of data that'
                              'follows last PCF parameter structure'
  Format       As String*8  'Format name of data that follows last PCF'
                              'parameter structure'
  Flags       As Long      'Flags'
  PCFHeader   As MQCFH     'Programmable command format header'
End Type

Global MQEPH_DEFAULT As MQEPH

```

### **StrucId (MQCHAR4) para MQEPH**

Este é o identificador da estrutura do cabeçalho PCF integrado. Ele é sempre um campo de entrada Seu valor é MQEPH\_STRUC\_ID.

O valor deve ser:.

#### **MQEPH\_STRUC\_ID**

Identificador para a estrutura do cabeçalho PCF integrado.

Para a linguagem de programação C, a constante MQEPH\_STRUC\_ID\_ARRAY também é definida.. Este tem o mesmo valor que MQEPH\_STRUC\_ID, mas é uma matriz de caracteres em vez de uma sequência.

### **Versão (MQLONG) para MQEPH**

O valor deve ser:.

#### **MQEPH\_VERSION\_1**

Número da versão para a estrutura do cabeçalho PCF integrado

A constante a seguir especifica o número da versão atual:

#### **MQCFH\_VERSION\_3**

A versão atual da estrutura do cabeçalho PCF integrado

O valor inicial desse campo é MQEPH\_VERSION\_1.

### **StrucLength (MQLONG) para MQEPH**

Esta é a quantidade de dados que precedem a próxima estrutura do cabeçalho... Ele inclui:

- O comprimento do cabeçalho MQEPH
- O comprimento de todos os parâmetros PCF após o cabeçalho
- Qualquer preenchimento em branco seguindo esses parâmetros

StrucLength deve ser um múltiplo de 4..

A parte de comprimento fixo da estrutura é definida por MQEPH\_STRUC\_LENGTH\_FIXED

O valor inicial desse campo é 68.

### **Codificação (MQLONG) para MQEPH**

Esta é a codificação numérica dos dados que seguem a estrutura MQEPH e os parâmetros PCF associados; ela não se aplica aos dados de caracteres na própria estrutura MQEPH

O valor inicial desse campo é 0.

### **CodedCharSetId (MQLONG) para MQEPH**

Esse é o identificador do conjunto de caracteres dos dados que seguem a estrutura MQEPH e os parâmetros PCF associados; ele não se aplica aos dados de caracteres na própria estrutura MQEPH

O valor inicial desse campo é MQCCSI\_UNDEFINED.

### **Formato (MQCHAR8) para MQEPH**

Este é o nome do formato dos dados que seguem a estrutura MQEPH e os parâmetros PCF associados

O valor inicial desse campo é MQFMT\_NONE.

### **Sinalizadores (MQLONG) para MQEPH**

Os seguintes valores estão disponíveis:

#### **MQEPH\_NONE**

Nenhuma sinalização foi especificada. MQEPH\_NONE é definido para a documentação do programa de auxílio Não se pretende que esta constante seja usada com qualquer outra, mas como seu valor é zero, tal uso não pode ser detectado.

#### **MQEPH\_CCSID\_EMBEDDED**

O conjunto de caracteres dos parâmetros que contêm dados de caractere é especificado individualmente no campo CodedCharSetId em cada estrutura. O conjunto de caracteres dos campos StrucId e Format é definido pelo campo CodedCharSetId na estrutura do cabeçalho que precede a estrutura MQEPH ou pelo campo CodedCharSetId no MQMD se o MQEPH estiver no início da mensagem.

O valor inicial desse campo é MQEPH\_NONE.

### **PCFHeader (MQCFH) para MQEPH**

Este é o cabeçalho PCF (Programmable Command Format), definindo os parâmetros PCF que seguem a estrutura MQEPH. Isso permite seguir os dados da mensagem PCF com outros cabeçalhos.

O cabeçalho PCF é inicialmente definido com os seguintes valores:

<i>Tabela 489. Campos em MQCFH</i>		
<b>Nome de Campo</b>	<b>Nome da constante</b>	<b>Valor da constante</b>
<i>Type</i>	MQCFT_NONE	0
<i>StrucLength</i>	MQCFH_STRUC_LENGTH	36
<i>Version</i>	MQCFH_VERSION_3	3
<i>StrucLength</i>	Nenhum	0
<i>Command</i>	MQCMD_NONE	0
<i>MsgSeqNumber</i>	Nenhum	1
<i>Control</i>	MQCFC_LAST	1
<i>CompCode</i>	MQCC_OK	0

Tabela 489. Campos em MQCFH (continuação)		
Nome de Campo	Nome da constante	Valor da constante
<i>Reason</i>	MQRC_NONE	0
<i>ParameterCount</i>	Nenhum	0

O aplicativo deve mudar o Type de MQCFT\_NONE para um tipo de estrutura válido para o uso que ele está fazendo do cabeçalho PCF integrado

## MQGMO-Obter opções de mensagem

A estrutura MQGMO permite que o aplicativo controle como as mensagens são removidas das filas. A estrutura é um parâmetro de entrada / saída na chamada MQGET..

### Versão

A versão atual de MQGMO é MQGMO\_VERSION\_4. Determinados campos estão disponíveis apenas em determinadas versões de MQGMO. Se você precisar portar aplicativos entre vários ambientes, deverá assegurar que a versão de MQGMO seja consistente em todos os ambientes. Os campos que existem apenas em versões específicas da estrutura são identificados como tal no [“MQGMO-Obter opções de mensagem”](#) na página 379 e nas descrições de campos.

Os arquivos de cabeçalho, COPY e INCLUDE fornecidos para as linguagens de programação suportadas contêm a versão mais recente de MQGMO suportada pelo ambiente, mas com o valor inicial do campo *Version* configurado como MQGMO\_VERSION\_1. Para usar campos que não estão presentes na estrutura version-1, configure o campo *Version* para o número da versão necessária.

### Conjunto de caracteres e codificação

Os dados em MQGMO devem estar no conjunto de caracteres fornecido pelo atributo do gerenciador de filas do **CodedCharSetId** e pela codificação do gerenciador de fila local fornecido por MQENC\_NATIVE. No entanto, se o aplicativo estiver em execução como um cliente MQI/MQ, a estrutura deverá estar no conjunto de caracteres e na codificação do cliente.

### Campos

**Nota:** Na tabela a seguir, os campos são agrupados por uso e não alfabeticamente. Os tópicos filhos seguem a mesma sequência.

Tabela 490. Campos em MQGMO para MQGMO		
Nome e descrição do campo	Nome da constante	Valor inicial (se houver) da constante
<u>StrucId</u> (identificador de estrutura)	MQGMO_STRUC_ID	'GMO-'
<u>Versão</u> (número da versão da estrutura).	MQGMO_VERSION_1	1
<u>MQGMO-Campo Opções</u> (opções que controlam a ação de MQGET)	MQGMO_NO_WAIT	0
<u>WaitInterval</u> (intervalo de espera)	Nenhum	0
<u>Signal1</u> (sinal)	Nenhum	Ponteiro nulo em z/OS ; 0 caso contrário
<u>Signal2</u> (identificador de sinal).	Nenhum	0
<u>ResolvedQName</u> (nome resolvido da fila de destino).	Nenhum	Cadeia nula ou espaços em branco

Tabela 490. Campos em MQGMO para MQGMO (continuação)

Nome e descrição do campo	Nome da constante	Valor inicial (se houver) da constante
<b>Nota:</b> Os campos restantes serão ignorados se <i>Version</i> for menor que MQGMO_VERSION_2.		
MatchOptions (opções que controlam critérios de seleção usados para MQGET)	MQMO_MATCH_MSG_ID + MQMO_MATCH_CORREL_ID	3
GroupStatus (sinalizador indicando se a mensagem recuperada está em um grupo)	MQGS_NOT_IN_GROUP	'␣'
SegmentStatus (sinalizador indicando se a mensagem recuperada é um segmento de uma mensagem lógica)	MQSS_NOT_A_SEGMENT	'␣'
Segmentação (sinalizador indicando se segmentação adicional é permitida para a mensagem recuperada)	MQSEG_INIBIDA	'␣'
Reserved1 (reservado)	Nenhum	'␣'
<b>Nota:</b> Os campos restantes serão ignorados se o <i>Version</i> for menor que MQGMO_VERSION_3		
MsgToken (token de mensagem);	MQMTOK_NONE	Nulos
ReturnedLength (comprimento em bytes de dados da mensagens retornados).	MQRL_UNDEFINED	-1
<b>Nota:</b> Os campos restantes serão ignorados se <i>Version</i> for menor que MQGMO_VERSION_4..		
Reserved2 (reservado)	Nenhum	'␣'
MsgHandle (identificador para uma mensagem que deve ser preenchida com as propriedades da mensagem que está sendo recuperada da fila)	MQHM_NONE	0
<p><b>Notas:</b></p> <ol style="list-style-type: none"> <li>1. O símbolo ␣ representa um único caractere em branco.</li> <li>2. O valor Sequência nula ou espaços em branco indica a sequência nula em C e caracteres em branco em outras linguagens de programação.</li> <li>3. Na linguagem de programação C, a variável macroMQGMO_DEFAULT contém os valores listados na tabela. Ele pode ser usado da seguinte maneira para fornecer valores iniciais para os campos na estrutura:</li> </ol> <pre>MQGMO MyGMO = {MQGMO_DEFAULT};</pre>		

## Declarações de idiomas

Declaração C para MQGMO

```
typedef struct tagMQGMO MQGMO;
struct tagMQGMO {
    MQCHAR4   StrucId;           /* Structure identifier */
    MQLONG    Version;          /* Structure version number */
    MQLONG    Options;          /* Options that control the action of */
                                /* MQGET */
};
```



```

MQLONG    WaitInterval;    /* Wait interval */
MQLONG    Signal1;         /* Signal */
MQLONG    Signal2;         /* Signal identifier */
MQCHAR48   ResolvedQName;  /* Resolved name of destination queue */
/* Ver:1 */
MQLONG    MatchOptions;    /* Options controlling selection */
/* criteria used for MQGET */
MQCHAR     GroupStatus;    /* Flag indicating whether message */
/* retrieved is in a group */
MQCHAR     SegmentStatus; /* Flag indicating whether message */
/* retrieved is a segment of a logical */
/* message */
MQCHAR     Segmentation;   /* Flag indicating whether further */
/* segmentation is allowed for the */
/* message retrieved */
MQCHAR     Reserved1;      /* Reserved */
/* Ver:2 */
MQBYTE16   MsgToken;       /* Message token */
MQLONG     ReturnedLength; /* Length of message data returned */
/* (bytes) */
/* Ver:3 */
MQLONG     Reserved2;      /* Reserved */
MQHMSG     MsgHandle;      /* Message handle */
/* Ver:4 */
};

```

**Nota:** No z/OS, o campo *Signal1* é declarado como PMQLONG

Declaração COBOL para MQGMO

```

** MQGMO structure
10 MQGMO.
** Structure identifier
15 MQGMO-STRUCID PIC X(4).
** Structure version number
15 MQGMO-VERSION PIC S9(9) BINARY.
** Options that control the action of MQGET
15 MQGMO-OPTIONS PIC S9(9) BINARY.
** Wait interval
15 MQGMO-WAITINTERVAL PIC S9(9) BINARY.
** Signal
15 MQGMO-SIGNAL1 PIC S9(9) BINARY.
** Signal identifier
15 MQGMO-SIGNAL2 PIC S9(9) BINARY.
** Resolved name of destination queue
15 MQGMO-RESOLVEDQNAME PIC X(48).
** Options controlling selection criteria used for MQGET
15 MQGMO-MATCHOPTIONS PIC S9(9) BINARY.
** Flag indicating whether message retrieved is in a group
15 MQGMO-GROUPSTATUS PIC X.
** Flag indicating whether message retrieved is a segment of a
** logical message
15 MQGMO-SEGMENTSTATUS PIC X.
** Flag indicating whether further segmentation is allowed for the
** message retrieved
15 MQGMO-SEGMENTATION PIC X.
** Reserved
15 MQGMO-RESERVED1 PIC X.
** Message token
15 MQGMO-MSGTOKEN PIC X(16).
** Length of message data returned (bytes)
15 MQGMO-RETURNEDLENGTH PIC S9(9) BINARY.
** Reserved
15 MQGMO-RESERVED2 PIC S9(9) BINARY.
** Message handle
15 MQGMO-MSGHANDLE PIC S9(18) BINARY.

```

**Nota:** No z/OS, o campo *Signal1* é declarado como POINTER

Declaração PL/I para MQGMO

```

dcl
1 MQGMO based,
3 StrucId char(4), /* Structure identifier */
3 Version fixed bin(31), /* Structure version number */
3 Options fixed bin(31), /* Options that control the action of
MQGET */

```

```

3 WaitInterval    fixed bin(31), /* Wait interval */
3 Signal1        fixed bin(31), /* Signal */
3 Signal2        fixed bin(31), /* Signal identifier */
3 ResolvedQName  char(48),      /* Resolved name of destination
                               queue */
3 MatchOptions   fixed bin(31), /* Options controlling selection
                               criteria used for MQGET */
3 GroupStatus     char(1),      /* Flag indicating whether message
                               retrieved is in a group */
3 SegmentStatus  char(1),      /* Flag indicating whether message
                               retrieved is a segment of a logical
                               message */
3 Segmentation   char(1),      /* Flag indicating whether further
                               segmentation is allowed for the
                               message retrieved */
3 Reserved1      char(1),      /* Reserved */
3 MsgToken       char(16),     /* Message token */
3 ReturnedLength fixed bin(31); /* Length of message data returned
                               (bytes) */
3 Reserved2      fixed bin(31); /* Reserved */
3 MsgHandle      fixed bin(63); /* Message handle */

```

**Nota:** No z/OS, o campo *Signal1* é declarado como pointer

#### Declaração High Level Assembler para MQGMO

```

MQGMO          DSECT
MQGMO_STRUCID  DS    CL4  Structure identifier
MQGMO_VERSION  DS    F    Structure version number
MQGMO_OPTIONS  DS    F    Options that control the action of
*              MQGET
MQGMO_WAITINTERVAL DS    F    Wait interval
MQGMO_SIGNAL1  DS    F    Signal
MQGMO_SIGNAL2  DS    F    Signal identifier
MQGMO_RESOLVEDQNAME DS    CL48 Resolved name of destination queue
MQGMO_MATCHOPTIONS DS    F    Options controlling selection criteria
*              used for MQGET
MQGMO_GROUPSTATUS DS    CL1  Flag indicating whether message
*              retrieved is in a group
MQGMO_SEGMENTSTATUS DS    CL1  Flag indicating whether message
*              retrieved is a segment of a logical
*              message
MQGMO_SEGMENTATION DS    CL1  Flag indicating whether further
*              segmentation is allowed for the message
*              retrieved
MQGMO_RESERVED1 DS    CL1  Reserved
MQGMO_MSGTOKEN  DS    XL16  Message token
MQGMO_RETURNEDLENGTH DS    F    Length of message data returned (bytes)
MQGMO_RESERVED2 DS    F    Reserved
MQGMO_MSGHANDLE DS    D    Message handle
MQGMO_LENGTH    EQU    *-MQGMO
                ORG    MQGMO
MQGMO_AREA      DS    CL(MQGMO_LENGTH)

```

#### Declaração High Level Assembler para MQGMO

```

Type MQGMO
  StrucId      As String*4  'Structure identifier'
  Version      As Long      'Structure version number'
  Options      As Long      'Options that control the action of MQGET'
  WaitInterval As Long      'Wait interval'
  Signal1      As Long      'Signal'
  Signal2      As Long      'Signal identifier'
  ResolvedQName As String*48 'Resolved name of destination queue'
  MatchOptions As Long      'Options controlling selection criteria'
  GroupStatus  As String*1  'Flag indicating whether message'
  SegmentStatus As String*1 'retrieved is in a group'
  Segmentation As String*1  'Flag indicating whether message'
  MsgToken     As MQBYTE16  'Message token'

```

ReturnedLength As Long  
End Type

'Length of message data returned (bytes)'

## PROPCTL opções de canal para MQGMO

Use o atributo do canal **PROPCTL** para controlar quais propriedades de mensagem são incluídas em uma mensagem que é enviada de um gerenciador de filas do IBM MQ 9.3 para um gerenciador de filas parceiro de uma versão anterior do IBM MQ.

Tabela 491. Configurações de Atributo de Propriedade de Mensagem do Canal

PROPCTL	Descrição
all	<p>Use essa opção se os aplicativos conectados ao gerenciador de fila parceiro de uma versão anterior forem capazes de processar quaisquer propriedades colocadas em uma mensagem por um aplicativo IBM MQ 9.3 .</p> <p>Todas as propriedades são enviadas para o gerenciador de filas parceiro, além de quaisquer pares nome-valor colocados no MQRFH2.</p> <p>Você deve levar em consideração dois problemas de design do aplicativo:</p> <ol style="list-style-type: none"><li>1. Um aplicativo conectado ao gerenciador de filas parceiro deve ser capaz de processar mensagens contendo cabeçalhos MQRFH2 gerados em um gerenciador de filas IBM MQ 9.3 .</li><li>2. O aplicativo conectado ao gerenciador de filas do parceiro deve processar novas propriedades de mensagem sinalizadas com MQPD_SUPPORT_REQUIRED corretamente.</li></ol> <p>Com a opção de canal ALL configurada, os aplicativos JMS podem interoperar entre IBM MQ 9.3 e uma versão anterior usando o canal. Novos aplicativos IBM MQ 9.3 que usam propriedades de mensagem podem interoperar com aplicativos de uma versão anterior, dependendo de como o aplicativo da versão anterior manipula cabeçalhos MQRFH2.</p>

Tabela 491. Configurações de Atributo de Propriedade de Mensagem do Canal (continuação)

PROPCTL	Descrição
COMPAT	<p>Use esta opção para enviar propriedades de mensagem aos aplicativos conectados a um gerenciador de filas de parceiro de versão anterior em alguns casos, mas não todos. Propriedades de mensagem só são enviadas se duas condições forem atendidas:</p> <ol style="list-style-type: none"> <li>1. Nenhuma propriedade deve ser marcada como requerendo processamento de propriedade de mensagem.</li> <li>2. Pelo menos uma das propriedades de mensagem deve ser configurada em uma pasta "reservada"; consulte <a href="#">Nota</a>.</li> </ol> <p>Com a opção de canal COMPAT configurada, os aplicativos JMS podem interoperar entre o IBM MQ 9.3 e uma versão anterior usando o canal.</p> <p>O canal não está disponível para todos os aplicativos que estão usando propriedades de mensagem, mas sim para aqueles que usam as pastas reservadas. As regras que mostram se a mensagem ou propriedade é enviada são:</p> <ol style="list-style-type: none"> <li>1. Se a mensagem tiver propriedades, mas nenhuma delas estiver associada a uma pasta "reservada", nenhuma propriedade de mensagem será enviada.</li> <li>2. Se alguma propriedade de mensagem tiver sido criada em uma pasta de propriedades "reservada", todas as propriedades de mensagem associadas à mensagem serão enviadas. Entretanto:             <ol style="list-style-type: none"> <li>a. Se alguma das propriedades de mensagem for marcada como suporte requerido, MQPD_SUPPORT_REQUIRED ou MQPD_SUPPORT_REQUIRED_IF_LOCAL, a mensagem inteira será rejeitada. Ela será retornada, descartada ou enviada para a fila de devoluções de acordo com o valor de suas opções de relatório.</li> <li>b. Se nenhuma propriedade de mensagem for marcada como suporte requerido, uma propriedade individual pode não ser enviada. Se algum dos campos de descritor da propriedade de mensagem for configurado como valores não padrão, a propriedade individual não será enviada. A mensagem ainda será enviada. Um exemplo de um valor do campo de descritor de propriedade não padrão é MQPD_USER_CONTEXT.</li> </ol> </li> </ol> <p><b>Nota:</b> Os nomes de pastas "reservados" começam com mcd., jms., usr. ou mqext.. Essas pastas são criadas para aplicativos que usam a interface do JMS. Em IBM MQ 9.3, quaisquer pares nome-valor colocados nessas pastas são tratados como propriedades de mensagem.</p> <p>As propriedades de mensagem são enviadas em um cabeçalho MQRFH2, além de quaisquer pares nome-valor colocados em um cabeçalho MQRFH2. Quaisquer pares nome-valor colocados em um cabeçalho MQRFH2 são enviados desde que a mensagem não seja rejeitada.</p>
NONE	<p>Use esta opção para evitar que qualquer propriedade de mensagem seja enviada aos aplicativos conectados a um gerenciador de filas de parceiro de versão anterior. Um MQRFH2 que contém pares nome-valor e propriedades de mensagem ainda é enviado, mas apenas com os pares nome-valor..</p> <p>Com a opção de canal NONE configurada, uma mensagem do JMS é enviada como JMSTextMessage ou JMSBytesMessage sem quaisquer propriedades de mensagem do JMS. Se for possível que um aplicativo de versão anterior ignore todas as propriedades configuradas no aplicativo IBM MQ 9.3, ele poderá interoperar com ele.</p>

### PROPCTL opções de fila para MQGMO

Use o atributo da fila **PROPCTL** para controlar como propriedades de mensagem são retornadas para um aplicativo que chama **MQGET** sem nenhuma opção de propriedade de mensagem **MQGMO**.

Tabela 492. Configurações de Atributo de Propriedade de Mensagem da Fila

PROPCTL	Descrição
all	<p>Use a opção ALL para que diferentes aplicativos que leem uma mensagem da mesma fila possam processar a mensagem de maneiras diferentes.</p> <ul style="list-style-type: none"> <li>• Um aplicativo, migrado inalterado a partir de uma versão anterior, pode continuar lendo o MQRFH2 diretamente. As propriedades ficam diretamente acessíveis no cabeçalho MQRFH2.</li> </ul> <p>Você deve modificar o aplicativo para manipular quaisquer novas propriedades e novos atributos de propriedade. É possível que o aplicativo seja afetado pelas mudanças no layout e no número de cabeçalhos MQRFH2. Alguns atributos de pasta podem ser removidos, ou aqueles que o IBM MQ relata um erro no layout do cabeçalho MQRFH2 que é ignorado em uma versão anterior.</p> <ul style="list-style-type: none"> <li>• Um aplicativo novo ou alterado pode usar o MQI de propriedade de mensagem para consultar propriedades de mensagens e ler pares nome-valor no cabeçalho MQRFH2 diretamente.</li> </ul> <p>Todas as propriedades na mensagem são retornadas para o aplicativo.</p> <ul style="list-style-type: none"> <li>• Se o aplicativo chamar MQCRTMH para criar uma manipulação de mensagem, ele deverá consultar as propriedades de mensagem usando MQINQMP. Os pares nome-valor que não são propriedades de mensagem permanecem no MQRFH2, que é removido de quaisquer propriedades de mensagens.</li> <li>• Se o aplicativo não criar uma manipulação de mensagens, todas as propriedades de mensagem e pares nome-valor permanecerão no MQRFH2.</li> </ul> <p>ALL só terá esse efeito se o aplicativo de recebimento não tiver configurado uma opção MQGMO_PROPERTIES ou o tiver configurado como MQGMO_PROPERTIES_AS_Q_DEF</p>

Tabela 492. Configurações de Atributo de Propriedade de Mensagem da Fila (continuação)

PROPCTL	Descrição
COMPAT (padrão)	<p>COMPAT é a opção padrão. Se <code>GM0_PROPERTIES_*</code> não estiver configurado como em um aplicativo não modificado a partir de uma versão anterior, COMPAT será assumido. Padronizando para a opção COMPAT, um aplicativo de versão anterior que não criou explicitamente um MQRFH2, funciona sem mudança no IBM MQ 9.3.</p> <p>Use esta opção se você tiver escrito um aplicativo MQI de aplicativo de versão anterior para ler mensagens do JMS.</p> <ul style="list-style-type: none"> <li>• As propriedades JMS, que são armazenadas em um cabeçalho MQRFH2, são retornadas para o aplicativo em um cabeçalho MQRFH2 em pastas com nomes que começam com <code>mc.</code>, <code>jms.</code> <code>usr.</code> ou <code>mqext.</code></li> <li>• Se a mensagem tiver pastas JMS e um aplicativo IBM MQ 9.3 incluir novas pastas de propriedades na mensagem, essas propriedades também serão retornadas no MQRFH2. Conseqüentemente, você deve modificar o aplicativo para manipular quaisquer novas propriedades e novos atributos de propriedade. É possível que um aplicativo seja afetado pelas mudanças no layout e no número de cabeçalhos MQRFH2. Ele pode descobrir que alguns atributos de pasta foram removidos, ou que o IBM MQ localiza erros no layout do cabeçalho MQRFH2 que é ignorado em uma versão anterior.</li> </ul> <p><b>Nota:</b> Neste cenário, o comportamento do aplicativo é o mesmo se ele estivesse conectado a uma versão ou gerenciador de filas do IBM MQ 9.3 anterior. Se o atributo <b>PROPCTL</b> do canal está configurado como COMPAT ou ALL, qualquer nova propriedade de mensagem é enviada na mensagem para o gerenciador de filas do parceiro de versão anterior.</p> <ul style="list-style-type: none"> <li>• Se a mensagem não for uma mensagem do JMS, mas contiver outras propriedades, essas propriedades não serão retornadas para o aplicativo em um cabeçalho MQRFH2.<sup>1</sup></li> <li>• A opção também permite que aplicativos de versão anterior que criam explicitamente um MQRFH2 funcionem corretamente, em muitos casos. Por exemplo, um programa MQI que cria um MQRFH2 contendo propriedades de mensagem do JMS continua funcionando corretamente. Se uma mensagem for criada sem propriedades de mensagem do JMS, mas com algumas outras pastas do MQRFH2, as pastas serão retornadas para o aplicativo. Somente se as pastas forem pastas de propriedades de mensagem essas pastas específicas serão removidas do MQRFH2. As pastas de propriedades de mensagem são identificadas por terem o novo atributo de pasta <code>content= 'properties'</code> ou por serem pastas com nomes listados em <u>Nome de Pasta de Propriedades Definido</u> ou <u>Nome de Pasta de Propriedades Não Agrupadas</u>.</li> <li>• Se o aplicativo chamar MQCRTMH para criar uma manipulação de mensagem, ele deverá consultar as propriedades de mensagem usando MQINQMP. Propriedades de mensagem são removidas dos cabeçalhos MQRFH2. Os pares nome-valor que não são propriedades de mensagem permanecem no MQRFH2.</li> <li>• Se o aplicativo chamar MQCRTMH para criar uma manipulação de mensagem, ele poderá consultar todas as propriedades de mensagem, independentemente se a mensagem tiver pastas do JMS.</li> <li>• Se o aplicativo não criar uma manipulação de mensagens, todas as propriedades de mensagem e pares nome-valor permanecerão no MQRFH2.</li> </ul> <p>Se uma mensagem contiver novas pastas de propriedades de usuários, é possível concluir que a mensagem foi criada por um aplicativo IBM MQ 9.3 novo ou alterado. Se o aplicativo de recebimento for processar essas novas propriedades diretamente em um MQRFH2, você deverá modificar o aplicativo para usar a opção ALL. Com a opção COMPAT padrão configurada, um aplicativo não modificado continua a processar o restante do MQRFH2, sem as propriedades da IBM MQ 9.3.</p> <p>A intenção da interface de PROPCTL é suportar aplicativos antigos que lêem pastas do MQRFH2 e aplicativos novos e alterados que usam a interface de propriedade de mensagem. Faça com que os novos aplicativos usem a interface de propriedade de mensagem para todas as propriedades de mensagem do usuário e evite ler e gravar cabeçalhos MQRFH2 diretamente.</p> <p>COMPAT é a opção padrão para aplicativos de recebimento. Se o atributo de canal PROPCTL está configurado como COMPAT ou ALL, qualquer nova propriedade de mensagem é enviada na mensagem para o gerenciador de filas do parceiro de versão anterior.</p>

Tabela 492. Configurações de Atributo de Propriedade de Mensagem da Fila (continuação)

PROPCTL	Descrição
FORCE	<p>A opção FORCE coloca todas as propriedades de mensagem nos cabeçalhos MQRFH2. Todas as propriedades de mensagem e pares nome-valor nos cabeçalho MQRFH2 permanecem na mensagem. As propriedades de mensagem não são removidas do MQRFH2 e disponibilizadas por meio de uma manipulação de mensagem. O efeito de escolher a opção FORCE é a ativação de um aplicativo recém-migrado para ler propriedades de mensagem de cabeçalhos MQRFH2.</p> <p>Suponha que você tenha modificado um aplicativo para processar propriedades de mensagens da IBM MQ 9.3, mas que também tenha mantido sua capacidade de trabalhar diretamente com cabeçalhos MQRFH2, como antes. É possível decidir quando alternar o aplicativo para usar propriedades de mensagem configurando inicialmente o atributo da fila PROPCTL para FORCE. Configure o atributo da fila <b>PROPCTL</b> para outro valor quando você estiver pronto para começar a usar propriedades de mensagem. Se a nova função no aplicativo não se comportar conforme esperado, configure a opção <b>PROPCTL</b> de volta para FORCE.</p> <p>FORCE só terá esse efeito se o aplicativo de recebimento não tiver configurado uma opção MQGMO_PROPERTIES ou tiver configurado como MQGMO_PROPERTIES_AS_Q_DEF.</p>
NONE	<p>Use a opção NONE para que um aplicativo existente possa processar uma mensagem, ignorando todas as propriedades de mensagem e um aplicativo novo ou alterado possa consultar propriedades de mensagem.</p> <ul style="list-style-type: none"> <li>• Se o aplicativo chamar MQCRTMH para criar uma manipulação de mensagem, ele deverá consultar as propriedades de mensagem usando MQINQMP. Os pares nome-valor que não são propriedades de mensagem permanecem no MQRFH2, que é removido de quaisquer propriedades de mensagens.</li> <li>• Se o aplicativo não criar uma manipulação de mensagem, todas as propriedades de mensagem serão removidas no MQRFH2. Os pares nome-valor nos cabeçalhos MQRFH2 permanecem na mensagem.</li> </ul> <p>NONE só terá esse efeito se o aplicativo de recebimento não tiver configurado uma opção MQGMO_PROPERTIES ou o tiver configurado como MQGMO_PROPERTIES_AS_Q_DEF</p>
V6COMPAT	<p>Use esta opção para receber um MQRFH2 no mesmo formato em que ele foi enviado. Se o aplicativo de envio ou gerenciador de filas, criar propriedades de mensagem adicionais, elas serão retornadas na manipulação de mensagem.</p> <p>Essa opção deve ser configurada nas filas de envio e de recebimento e em quaisquer filas de transmissão intervenientes. Ela substitui quaisquer outras opções PROPCTL configuradas em definições de fila no caminho de resolução de nome de fila.</p> <p>Use a opção V6COMPAT apenas em circunstâncias excepcionais. Por exemplo, se você estiver migrando aplicativos de uma versão anterior para IBM MQ 9.3, a opção será valiosa porque preserva o comportamento da versão anterior. É provável que a opção tenha um impacto no rendimento da mensagem. E ela também é mais difícil de administrar; é preciso assegurar que a opção esteja configurada nas filas do emissor, do receptor e de transmissão intervenientes.</p> <p>V6COMPAT só terá esse efeito se o aplicativo de recebimento não tiver configurado uma opção MQGMO_PROPERTIES ou tiver configurado como MQGMO_PROPERTIES_AS_Q_DEF.</p>

<sup>1</sup> A existência de pastas de propriedade específicas criadas pelo IBM MQ classes for JMS indica uma mensagem JMS As pastas de propriedades são mcd., jms.usr. ou mqext

Para obter mais informações sobre propriedades de mensagem e pares nome-valor, consulte [“NameValueData \(MQCHARn\) para MQRFH2”](#) na página 553.

## Opções de propriedade de mensagem para MQGMO ..

Use as opções da propriedade de mensagens **MQGMO** para controlar como as propriedades da mensagem são retornadas para um aplicativo

<i>Tabela 493. Configurações de Opções da Propriedade de Mensagem MQGMO</i>	
<b>Opção MQGMO</b>	<b>Descrição</b>
MQGMO_PROPERTIES_AS_Q_DEF	<p>Os aplicativos IBM MQ que lêem a partir da mesma fila e não configuram <code>GMO_PROPERTIES_*</code>, recebem as propriedades de mensagem de forma diferente. Os aplicativos IBM MQ que não criam uma manipulação de mensagem são controlados pelo atributo <b>PROPCTL</b> da fila. Um aplicativo IBM MQ pode optar por receber propriedades de mensagens no MQRFH2 ou criar uma manipulação de mensagens e consultar as propriedades de mensagens. Se o aplicativo criar uma manipulação de mensagem, as propriedades serão removidas do MQRFH2.</p> <ul style="list-style-type: none"> <li>• Um aplicativo IBM MQ novo ou alterado que não configura <code>GMO_PROPERTIES_*</code> ou que a configura como <code>MQGMO_PROPERTIES_AS_Q_DEF</code> pode optar por consultar propriedades de mensagem. Ele deve configurar <code>MQCRTMH</code> para criar uma manipulação de mensagem e consultar propriedades de mensagem usando a chamada <code>MQI MQINQMP</code>.</li> <li>• Se um aplicativo novo ou alterado não criar uma manipulação de mensagens, ele deverá ler quaisquer propriedades de mensagens que ele receber diretamente dos cabeçalhos MQRFH2.</li> <li>• Se o atributo da fila <b>PROPCTL</b> estiver configurado como <code>FORCE</code>, nenhuma propriedade será retornada na manipulação de mensagem. Todas as propriedades são retornadas nos cabeçalhos MQRFH2.</li> <li>• Se o atributo da fila <b>PROPCTL</b> for configurado como <code>NONE</code> ou <code>COMPAT</code>, um aplicativo IBM MQ que cria uma manipulação de mensagens receberá todas as propriedades de mensagens.</li> </ul>
MQGMO_PROPERTIES_IN_HANDLE	<p>Força um aplicativo a usar propriedades de mensagem. Use esta opção para detectar se um aplicativo modificado falha ao criar uma manipulação de mensagem. O aplicativo pode estar tentando ler propriedades de mensagem diretamente de um MQRFH2, e não da chamada <code>MQI MQINQMP</code>.</p>
MQGMO_NO_PROPERTIES	<ul style="list-style-type: none"> <li>• Todas as propriedades são removidas. Propriedades geradas pelo gerenciador de filas, como <code>JMS</code>, são removidas.</li> <li>• Propriedades são removidas mesmo se uma manipulação de mensagem for criada. Pares nome-valor em outras pastas MQRFH2 estão disponíveis nos dados da mensagem.</li> </ul>



Tabela 493. Configurações de Opções da Propriedade de Mensagem MQGMO (continuação)

Opção MQGMO	Descrição
MQGMO_PROPERTIES_FORCE_MQRFH2	<p>Propriedades são retornadas nos cabeçalhos MQRFH2, mesmo se uma manipulação de mensagem for criada.</p> <ul style="list-style-type: none"> <li>MQINQMP não retorna propriedades de mensagem, mesmo se uma manipulação de mensagem for criada. MQRC_PROPERTY_NOT_AVAILABLE é retornado se uma propriedade for questionada.</li> </ul>
MQGMO_PROPERTIES_COMPATIBILITY	<p>Se a mensagem for de um cliente JMS, as propriedades JMS serão retornadas nos cabeçalhos MQRFH2. Aplicativos IBM MQ novos ou modificados, que criam uma manipulação de mensagem, se comportam de forma diferente.</p> <ul style="list-style-type: none"> <li>Todas as propriedades em quaisquer pastas de propriedades de mensagem são retornadas se a mensagem contiver uma pasta mcd., jms.usr. ou mqext.</li> <li>Se a mensagem contiver pastas de propriedades, mas não uma pasta mcd., jms.usr. ou mqext, nenhuma propriedade de mensagem será retornada em um MQRFH2.</li> <li>Se um aplicativo IBM MQ novo ou modificado criar uma manipulação de mensagem, consulte as propriedades de mensagem usando a chamada MQI MQINQMP. Todas as propriedades de mensagem são removidas do MQRFH2.</li> <li>Se um aplicativo IBM MQ novo ou modificado criar uma manipulação de mensagem, todas as propriedades na mensagem poderão ser consultadas. Mesmo se a mensagem não contiver uma pasta mcd., jms.usr. ou mqext, todas as propriedades de mensagem serão consultáveis.</li> </ul>

#### Referências relacionadas

PROPCTL

2471 (09A7) (RC2471): MQRC\_PROPERTY\_NOT\_AVAILABLE

#### **StrucId (MQCHAR4) para MQGMO**

Este é o identificador de estrutura da estrutura obter opções de mensagem. Ele é sempre um campo de entrada Seu valor é MQGMO\_STRUC\_ID.

O valor deve ser:.

#### **MQGMO\_STRUC\_ID**

Identificador para a estrutura obter opções de mensagem.

Para a linguagem de programação C, a constante MQGMO\_STRUC\_ID\_ARRAY também é definida. Ele tem o mesmo valor que MQGMO\_STRUC\_ID, mas é uma matriz de caracteres em vez de uma sequência.

#### **Versão (MQLONG) para MQGMO**

A versão é o número da versão da estrutura

O valor deve ser um dos seguintes:

#### **MQGMO\_VERSION\_1**

Version-1 estrutura de opções get-message.

Esta versão é suportada em todos os ambientes

## **MQGMO\_VERSION\_2**

Version-2 -estrutura de opções get-message.

Esta versão é suportada em todos os ambientes

## **MQGMO\_VERSION\_3**

Version-3 estrutura de opções get-message.

Esta versão é suportada em todos os ambientes

## **MQGMO\_VERSION\_4**

Version-4 estrutura de opções get-message.

Esta versão é suportada em todos os ambientes

Os campos que existem apenas nas versões mais recentes da estrutura são identificados como tal nas descrições dos campos A constante a seguir especifica o número da versão atual:

## **MQGMO\_CURRENT\_VERSION**

A versão atual da estrutura de opções get-message

Isto é sempre um campo de entrada. O valor inicial desse campo é MQGMO\_VERSION\_1.

## **Opções (MQLONG) para MQGMO**

**MQGMO** opções controlam a ação de MQGET. É possível especificar zero ou mais opções. Se você precisar de mais de um valor opcional:

- Inclua os valores (não inclua a mesma constante mais de uma vez) ou
- Combine os valores usando a operação OR bit a bit (se a linguagem de programação suportar operações de bit).

Combinações de opções inválidas são observadas; todas as outras combinações são válidas.

## **Opções de espera..**

As opções a seguir estão relacionadas à espera de mensagens chegarem na fila:

### **MQGMO\_WAIT**

O aplicativo aguarda até a chegada de uma mensagem adequada O tempo máximo que o aplicativo aguarda é especificado em *WaitInterval*..

**Importante:** Não há espera, ou atraso, se uma mensagem adequada estiver disponível imediatamente.

Se as solicitações MQGET forem inibidas ou as solicitações MQGET forem inibidas durante a espera, a espera será cancelada.. A chamada é concluída com MQCC\_FAILED e código de razão MQRC\_GET\_INHIBITED, independentemente de haver mensagens adequadas na fila.

É possível usar o MQGMO\_WAIT com as opções MQGMO\_BROWSE\_FIRST ou MQGMO\_BROWSE\_NEXT

Se vários aplicativos estiverem aguardando na mesma fila compartilhada, as regras a seguir selecionam qual aplicativo será ativado quando uma mensagem adequada chegar:

<i>Tabela 494. Regras para ativar chamadas MQGET em uma fila compartilhada.</i>		
<b>Número de chamadas MQGET aguardando para serem ativadas</b>		<b>Resultado</b>
<b>Com uma opção BROWSE</b>	<b>Sem uma opção BROWSE<sup>2</sup></b>	
Nenhum	um ou mais	Uma chamada MQGET sem uma opção BROWSE está ativada..

<sup>2</sup> Uma chamada MQGET que especifica a opção MQGMO\_LOCK é tratada como uma chamada não de procura.

Tabela 494. Regras para ativar chamadas MQGET em uma fila compartilhada. (continuação)

Número de chamadas MQGET aguardando para serem ativadas		Resultado
Com uma opção BROWSE	Sem uma opção BROWSE <sup>2</sup>	
um ou mais	Nenhum	Todas as chamadas MQGET com uma opção BROWSE são ativadas..
um ou mais	um ou mais	Uma chamada MQGET sem uma opção BROWSE está ativada.. O número de chamadas MQGET com uma opção BROWSE ativada é imprevisível.

Se mais de uma chamada MQGET sem uma opção BROWSE estiver aguardando na mesma fila, apenas uma será ativada. O gerenciador de filas tenta dar prioridade às chamadas em espera na seguinte ordem:

1. Solicitações get-wait específicas que podem ser atendidas apenas por determinadas mensagens, por exemplo, aquelas com um `MsgId` ou `CorrelId` específico (ou ambos).
2. Solicitações get-wait gerais que podem ser satisfeitas por qualquer mensagem.

**Nota:**

- Dentro da primeira categoria, nenhuma prioridade adicional é dada a solicitações get-wait mais específicas. Por exemplo, solicitações que especificam `MsgId` e `CorrelId`.
- Em qualquer categoria, não é possível prever qual aplicativo está selecionado. Em particular, o aplicativo que espera mais tempo não é necessariamente aquele selecionado.
- O comprimento do caminho e as considerações de planejamento de prioridade do sistema operacional podem significar que um aplicativo em espera de prioridade do sistema operacional inferior ao esperado recupera a mensagem.
- Também pode acontecer que um aplicativo que não está esperando recupere a mensagem em preferência a uma que esteja..

 Em z/OS, aplicam-se os seguintes pontos:

- Se desejar que o aplicativo continue com outro trabalho enquanto aguarda a chegada da mensagem, considere usar a opção de sinal (`MQGMO_SET_SIGNAL`) no lugar. No entanto, a opção de sinal é específica do ambiente; os aplicativos que você porta entre diferentes ambientes não devem usá-lo.
- Se houver mais de uma chamada MQGET aguardando a mesma mensagem, com uma mistura de opções de espera e de sinal, cada chamada de espera será considerada igualmente. É um erro especificar `MQGMO_SET_SIGNAL` com `MQGMO_WAIT`.. Também é um erro especificar essa opção com um identificador de fila para o qual um sinal está pendente.
- Se você especificar `MQGMO_WAIT` ou `MQGMO_SET_SIGNAL` para uma fila que tenha um `IndexType` de `MQIT_MSG_TOKEN`, nenhum critério de seleção será permitido. Isto significa que:
  - Se estiver usando um version-1 MQGMO, configure os campos `MsgId` e `CorrelId` no MQMD especificado na chamada MQGET para `MQMI_NONE` e `MQCI_NONE`.
  - Se estiver usando um version-2 ou posterior MQGMO, configure o campo `MatchOptions` como `MQMO_NONE`.
- Para uma chamada MQGET em uma fila compartilhada e a chamada é uma solicitação de procura ou um get destrutivo de uma mensagem de grupo, e nem `MsgId` nem `CorrelId` devem ser correspondidos, seu sinal ECB é postado `MQEC_MSG_CHEGOU` após 200 milissegundos.

<sup>2</sup> Uma chamada MQGET que especifica a opção `MQGMO_LOCK` é tratada como uma chamada não de procura.

<sup>2</sup> Uma chamada MQGET que especifica a opção `MQGMO_LOCK` é tratada como uma chamada não de procura.

Isso ocorre, mesmo que uma mensagem adequada possa não ter chegado na fila, até que o intervalo de espera tenha expirado, quando a fila for postada com MQEC\_WAIT\_INTERVAL\_EXPIRED. Quando MQEC\_MSG\_CHEGOU é postada, deve-se emitir novamente uma segunda chamada MQGET para recuperar a mensagem, se uma estiver disponível

Essa técnica é usada para assegurar que você seja informado em tempo hábil de uma chegada de mensagem, mas pode aparecer como uma sobrecarga de processamento inesperada quando comparada com uma seqüência de chamada semelhante em uma fila não compartilhada

MQGMO\_WAIT será ignorado se especificado com MQGMO\_BROWSE\_MSG\_UNDER\_CURSOR ou MQGMO\_MSG\_UNDER\_CURSOR ; Nenhum erro é emitido

### **MQGMO\_NO\_WAIT**

O aplicativo não esperará se nenhuma mensagem adequada estiver disponível MQGMO\_NO\_WAIT é o oposto do MQGMO\_WAIT.. MQGMO\_NO\_WAIT é definido para a documentação do programa de auxílio Ele será o padrão se nenhum for especificado

### **MQGMO\_SET\_SIGNAL**

Use essa opção com os campos Signal1 e Signal2 .. Ele permite que os aplicativos continuem com outro trabalho enquanto aguardam uma mensagem chegar. Ele também permite (se os recursos adequados do sistema operacional estiverem disponíveis) que os aplicativos aguardem mensagens que chegam em mais de uma fila.

**Nota:** A opção MQGMO\_SET\_SIGNAL é específica do ambiente; não a use para aplicativos que você deseja portar.

Em duas circunstâncias, a chamada é concluída da mesma maneira que se essa opção não tivesse sido especificada:

1. Se uma mensagem atualmente disponível satisfizer os critérios especificados no descritor de mensagens.
2. Se um erro de parâmetro ou outro erro síncrono for detectado,

Se nenhuma mensagem que satisfaça os critérios especificados no descritor de mensagens estiver atualmente disponível, o controle retornará ao aplicativo sem esperar que uma mensagem chegue. Os parâmetros **CompCode** e **Reason** são configurados como MQCC\_WARNING e MQRC\_SIGNAL\_REQUEST\_ACCEPTED Outros campos de saída no descritor de mensagens e os parâmetros de saída da chamada MQGET não são configurados Quando uma mensagem adequada chega mais tarde, o sinal é entregue através do envio do BCE.

O responsável pela chamada deve então emitir novamente a chamada MQGET para recuperar a mensagem. A aplicação pode esperar por este sinal, usando funções fornecidas pelo sistema operacional.

Se o sistema operacional fornecer um mecanismo de espera múltiplo, será possível usá-lo para aguardar uma mensagem chegar em qualquer uma das várias filas..

Se um WaitInterval diferente de zero for especificado, o sinal será entregue após o intervalo de espera expirar. O gerenciador de filas também pode cancelar a espera, nesse caso, o sinal é entregue

Mais de uma chamada de MQGET pode configurar um sinal para a mesma mensagem A ordem na qual os aplicativos são ativados é a mesma descrita para MQGMO\_WAIT.

Se mais de uma chamada MQGET estiver aguardando a mesma mensagem, cada chamada em espera será considerada igualmente As chamadas podem incluir uma combinação de opções de espera e sinal.

Sob determinadas condições, a chamada MQGET pode recuperar uma mensagem e um sinal resultante da chegada da mesma mensagem pode ser entregue.. Quando um sinal é entregue, um aplicativo deve ser preparado para que nenhuma mensagem esteja disponível;

Um identificador de fila não pode ter mais do que um pedido de sinal pendente

Essa opção não é válida com nenhuma das opções a seguir:

- MQGMO\_UNLOCK

- MQGMO\_WAIT

Para uma chamada MQGET em uma fila compartilhada e a chamada é uma solicitação de procura ou um get destrutivo de uma mensagem de grupo, e nem `MsgId` nem `CorrelId` devem ser correspondidos, o sinal do usuário ECB é postado MQEC\_MSG\_ARRIVED após 200 milissegundos.

Isso ocorre, embora uma mensagem adequada possa não ter chegado na fila, até que o intervalo de espera tenha expirado, quando a fila é postada com MQEC\_WAIT\_INTERVAL\_EXPIRED. Quando MQEC\_MSG\_ARRIVED é postado, deve-se emitir novamente uma segunda chamada MQGET para recuperar a mensagem, se uma estiver disponível

Essa técnica é usada para assegurar que você seja informado em tempo hábil de uma chegada de mensagem, mas pode aparecer como uma sobrecarga de processamento inesperada quando comparada com uma seqüência de chamada semelhante em uma fila não compartilhada

Este não é um método eficiente de recuperação de mensagens quando as mensagens são incluídas com pouca freqüência Para evitar essa sobrecarga para o caso de procura, especifique `MsgId` (se não indexado ou indexado por `MsgId`) ou `CorrelId` (se indexado por `CorrelId`) correspondente na chamada MQGET .

**z/OS** Essa opção é suportada somente no z/OS

### MQGMO\_FAIL\_IF QUIESCING

Force a chamada MQGET a falhar se o gerenciador de filas estiver no estado de quiesce

**z/OS** No z/OS, essa opção também força a chamada MQGET a falhar se a conexão (para um aplicativo CICS ou IMS ) estiver no estado de quiesce.

Se essa opção for especificada com MQGMO\_WAIT ou MQGMO\_SET\_SIGNALE a espera ou o sinal estiver pendente no momento em que o gerenciador de filas entrar no estado de quiesce:

- A espera é cancelado e a chamada retorna o código de conclusão MQCC\_FAILED com código de razão MQRC\_Q\_MGR QUIESCING ou MQRC\_CONNECTION QUIESCING.
- O sinal é cancelada com um código de conclusão de sinal específico do ambiente.

**z/OS** No z/OS, o sinal é concluído com o código de finalização de evento MQEC\_Q\_MGR QUIESCING ou MQEC\_CONNECTION QUIESCING

Se MQGMO\_FAIL\_IF QUIESCING não for especificado e o gerenciador de filas ou a conexão entrar no estado de quiesce, a espera ou o sinal não será cancelada

## Opções do ponto de sincronização

As opções a seguir estão relacionadas à participação da chamada MQGET em uma unidade de trabalho:

### MQGMO\_SYNCPOINT

A solicitação é operar dentro dos protocolos normais de unidade de trabalho. A mensagem é marcada como estando indisponível para outros aplicativos, mas é excluída da fila somente quando a unidade de trabalho é confirmada A mensagem é disponibilizada novamente se a unidade de trabalho for recuperada.

É possível deixar MQGMO\_SYNCPOINT e MQGMO\_NO\_SYNCPOINT desconfigurado Nesse caso, a inclusão da solicitação get em protocolos de unidade de trabalho é determinada pelo ambiente executando o gerenciador de filas. Ele não é determinado pelo ambiente executando o aplicativo.

- **z/OS** No z/OS, a solicitação get está em uma unidade de trabalho.
- Em todos os ambientes, exceto z/OS, a solicitação get não está em uma unidade de trabalho.

Devido a essas diferenças, um aplicativo que você deseja portar não deve permitir que essa opção seja padrão; especifique MQGMO\_SYNCPOINT ou MQGMO\_NO\_SYNCPOINT explicitamente.

Essa opção não é válida com nenhuma das opções a seguir:

- MQGMO\_BROWSE\_FIRST

- MQGMO\_BROWSE\_MSG\_UNDER\_CURSOR
- MQGMO\_BROWSE\_NEXT
- MQGMO\_LOCK
- MQGMO\_NO\_SYNCPOINT
- MQGMO\_SYNCPOINT\_IF\_PERSISTENT
- MQGMO\_UNLOCK

### **MQGMO\_SYNCPOINT\_IF\_PERSISTENT**

A solicitação é operar dentro dos protocolos normais de unidade de trabalho, mas apenas se a mensagem recuperada for persistente. Uma mensagem persistente tem o valor MQPER\_PERSISTENT no campo Persistence em MQMD.

- Se a mensagem for persistente, o gerenciador de filas processará a chamada como se o aplicativo tivesse especificado MQGMO\_SYNCPOINT
- Se a mensagem não for persistente, o gerenciador de filas processará a chamada como se o aplicativo tivesse especificado MQGMO\_NO\_SYNCPOINT

Essa opção não é válida com nenhuma das opções a seguir:

- MQGMO\_BROWSE\_FIRST
- MQGMO\_BROWSE\_MSG\_UNDER\_CURSOR
- MQGMO\_BROWSE\_NEXT
- MQGMO\_COMPLETE\_MSG
- MQGMO\_MARK\_SKIP\_BACKOUT
- MQGMO\_NO\_SYNCPOINT
- MQGMO\_SYNCPOINT
- MQGMO\_UNLOCK

Essa opção é suportada nos seguintes ambientes:

-  AIX
-  IBM i
-  Linux
-  z/OS


e para o IBM MQ MQI clients conectado a esses sistemas

### **MQGMO\_NO\_SYNCPOINT**

O pedido é para operar fora dos protocolos normais de unidade de trabalho. Se você obtiver uma mensagem sem uma opção de navegação, ela será excluída da fila imediatamente. A mensagem não pode ser disponibilizada novamente, efetuando backout da unidade de trabalho.

Essa opção será assumida se você especificar MQGMO\_BROWSE\_FIRST ou MQGMO\_BROWSE\_NEXT.

É possível deixar MQGMO\_SYNCPOINT e MQGMO\_NO\_SYNCPOINT desconfigurado. Nesse caso, a inclusão da solicitação get em protocolos de unidade de trabalho é determinada pelo ambiente executando o gerenciador de filas. Ele não é determinado pelo ambiente executando o aplicativo.

-  z/OS No z/OS, a solicitação get está em uma unidade de trabalho.
- Em todos os ambientes, exceto z/OS, a solicitação get não está em uma unidade de trabalho.

Devido a essas diferenças, um aplicativo que você deseja portar não deve permitir que essa opção seja padrão; especifique MQGMO\_SYNCPOINT ou MQGMO\_NO\_SYNCPOINT explicitamente.

Essa opção não é válida com nenhuma das opções a seguir:

- MQGMO\_MARK\_SKIP\_BACKOUT

- MQGMO\_SYNCPOINT
- MQGMO\_SYNCPOINT\_IF\_PERSISTENT

### **MQGMO\_MARK\_SKIP\_BACKOUT**

Retorne uma unidade de trabalho sem restabelecer na fila a mensagem que foi marcada com essa opção

Essa opção é suportada apenas no z/OS

Se essa opção for especificada, MQGMO\_SYNCPOINT também deverá ser especificado..  
MQGMO\_MARK\_SKIP\_BACKOUT não é válido com nenhuma das opções a seguir:

- MQGMO\_BROWSE\_FIRST
- MQGMO\_BROWSE\_MSG\_UNDER\_CURSOR
- MQGMO\_BROWSE\_NEXT
- MQGMO\_LOCK
- MQGMO\_NO\_SYNCPOINT
- MQGMO\_SYNCPOINT\_IF\_PERSISTENT
- MQGMO\_UNLOCK

**Nota:** Em IMS e CICS, você pode ter que emitir uma chamada IBM MQ extran depois de efetuar backout de uma unidade de trabalho contendo uma mensagem marcada com MQGMO\_MARK\_SKIP\_BACKOUT. Você deve emitir uma chamada IBM MQ antes de confirmar a nova unidade de trabalho contendo a mensagem marcada. A chamada pode ser qualquer chamada IBM MQ que você gosta.

1. No IMS, se você não tiver aplicado IMS APAR PN60855 e estiver executando um aplicativo IMS MPP ou BMP.
2. No CICS, se você estiver executando qualquer aplicativo,

Em ambos os casos, emita qualquer chamada IBM MQ antes de confirmar a nova unidade de trabalho que contém a mensagem restaurada.

**Nota:** Em uma unidade de trabalho, pode haver apenas uma solicitação get marcada como ignorando restauração, bem como nenhuma ou várias solicitações get não marcadas.

Se um aplicativo sair de uma unidade de trabalho, uma mensagem que foi recuperada usando MQGMO\_MARK\_SKIP\_BACKOUT não será restaurada para seu estado anterior. Outras atualizações de recursos são restaurados.... A mensagem é tratada como se fosse recuperada em uma nova unidade de trabalho iniciada pela solicitação de restauração. A mensagem é recuperada sem a opção MQGMO\_MARK\_SKIP\_BACKOUT ..

MQGMO\_MARK\_SKIP\_BACKOUT será útil se, após alguns recursos terem sido mudados, se tornar aparente que a unidade de trabalho não pode ser concluída com êxito Se você omitir essa opção, restaurar a unidade de trabalho restabelecerá a mensagem na fila. A mesma sequência de eventos ocorre novamente, quando a mensagem é recuperada em seguida.

No entanto, se você especificar MQGMO\_MARK\_SKIP\_BACKOUT na chamada MQGET original, a restauração da unidade de trabalho volta as atualizações para os outros recursos. A mensagem é tratada como se fosse recuperada sob uma nova unidade de trabalho. O aplicativo pode executar a manipulação de erros apropriada Ele pode enviar uma mensagem de relatório para o remetente da mensagem original ou colocar a mensagem original na fila de mensagens não entregues Ele pode então confirmar a nova unidade de trabalho. A confirmação da nova unidade de trabalho remove a mensagem permanentemente da fila original

MQGMO\_MARK\_SKIP\_BACKOUT marca uma única mensagem física.. Se a mensagem pertencer a um grupo de mensagens, as outras mensagens no grupo não serão marcados Da mesma forma, se a mensagem marcada for um segmento de uma mensagem lógica, os outros segmentos na mensagem lógica não serão marcados..

Qualquer mensagem em um grupo pode ser marcada, mas se mensagens forem recuperadas usando MQGMO\_LOGICAL\_ORDER, será vantajoso marcar a primeira mensagem no grupo. Se a unidade de trabalho for restaurada, a primeira mensagem (marcada) será movida para a nova unidade de trabalho. A segunda e mais recente mensagens no grupo são restabelecidas na fila. As mensagens restantes na fila não podem ser recuperadas por outro aplicativo usando MQGMO\_LOGICAL\_ORDER. A primeira mensagem no grupo não está mais na fila.. No entanto, o aplicativo que fez backup da unidade de trabalho pode recuperar as mensagens da segunda e mais recente na nova unidade de trabalho usando a opção MQGMO\_LOGICAL\_ORDER. A primeira mensagem já foi recuperada..

Ocasionalmente, pode ser necessário voltar para a nova unidade de trabalho. Por exemplo, porque a fila de mensagens não entregues está cheia, e a mensagem não deve ser descartada. A restauração da nova unidade de trabalho restabelece a mensagem na fila original, o que impede que a mensagem seja perdida. No entanto, nesta situação, o processamento não pode continuar. Após a restauração da nova unidade de trabalho, o aplicativo deve informar ao operador ou administrador que há um erro irrecuperável e, em seguida, concluir.

MQGMO\_MARK\_SKIP\_BACKOUT só funcionará se a unidade de trabalho que contém a solicitação get for interrompida pelo aplicativo que está fazendo backup dela. Se a unidade de trabalho que contém a solicitação de obtenção for restaurada porque a transação ou o sistema falhou, MQGMO\_MARK\_SKIP\_BACKOUT será ignorado. Qualquer mensagem recuperada usando esta opção é restabelecida na fila da mesma maneira que as mensagens recuperadas sem esta opção.

## Procurar Opções

As opções a seguir estão relacionadas à procura de mensagens na fila:

### MQGMO\_BROWSE\_FIRST

Quando uma fila for aberta com a opção MQOO\_BROWSE, um cursor de procura será estabelecido, posicionado logicamente antes da primeira mensagem na fila. Em seguida, é possível usar chamadas MQGET especificando a opção MQGMO\_BROWSE\_FIRST, MQGMO\_BROWSE\_NEXT ou MQGMO\_BROWSE\_MSG\_UNDER\_CURSOR para recuperar mensagens da fila de forma não destrutiva. O cursor de navegação marca a posição, dentro das mensagens na fila, a partir da qual a próxima chamada MQGET com MQGMO\_BROWSE\_NEXT procura uma mensagem adequada.

MQGMO\_BROWSE\_FIRST não é válido com nenhuma das opções a seguir:

- MQGMO\_BROWSE\_MSG\_UNDER\_CURSOR
- MQGMO\_BROWSE\_NEXT
- MQGMO\_MARK\_SKIP\_BACKOUT
- MQGMO\_MSG\_UNDER\_CURSOR
- MQGMO\_SYNCPOINT
- MQGMO\_SYNCPOINT\_IF\_PERSISTENT
- MQGMO\_UNLOCK

Também será um erro se a fila não tiver sido aberta para procura.

Uma chamada MQGET com MQGMO\_BROWSE\_FIRST ignora a posição anterior do cursor. A primeira mensagem na fila que satisfaça as condições especificadas no descritor de mensagens é recuperada. A mensagem permanece na fila e o cursor de navegação está posicionado nesta mensagem.

Após essa chamada, o cursor de navegação é posicionado sobre a mensagem que foi retornada. A mensagem pode ser removida da fila antes da próxima chamada MQGET com MQGMO\_BROWSE\_NEXT ser emitida. Nesse caso, o cursor de navegação permanece na posição na fila que a mensagem ocupou, mesmo que essa posição esteja vazia.

Use a opção MQGMO\_MSG\_UNDER\_CURSOR com uma chamada MQGET de não navegação para remover a mensagem da fila.

O cursor de navegação não é movido por uma chamada MQGET não de navegação, mesmo se estiver usando a mesma manipulação *Hobj*. Também não é movido por uma chamada MQGET



de navegação que retorna um código de conclusão de MQCC\_FAILED ou um código de razão de MQRC\_TRUNCATED\_MSG\_FAILED

Especifique a opção MQGMO\_LOCK com essa opção para bloquear a mensagem procurada.

É possível especificar MQGMO\_BROWSE\_FIRST com qualquer combinação válida das opções MQGMO\_\* e MQMO\_\* que controlam o processamento das mensagens em grupos e segmentos de mensagens lógicas.

Se você especificar MQGMO\_LOGICAL\_ORDER, as mensagens serão procuradas em ordem lógica. Se você omitir essa opção, as mensagens serão procuradas em ordem física. Se você especificar MQGMO\_BROWSE\_FIRST, será possível alternar entre ordem lógica e ordem física. Chamadas MQGET subsequentes usando MQGMO\_BROWSE\_NEXT navegam na fila na mesma ordem que a chamada mais recente que especificou MQGMO\_BROWSE\_FIRST para o identificador de filas.

O gerenciador de filas retém dois conjuntos de informações de grupo e segmento para chamadas MQGET. As informações de grupo e de segmento para chamadas de procura são retidas separadamente das informações para chamadas que removem mensagens da fila. Se você especificar MQGMO\_BROWSE\_FIRST, o gerenciador de filas ignorará as informações do grupo e do segmento para navegação. Ele varre a fila como se não houvesse grupo atual e nenhuma mensagem lógica atual. Se a chamada MQGET for bem-sucedida, o código de conclusão MQCC\_OK ou MQCC\_WARNING, as informações de grupo e segmento para navegação serão configuradas para aquela da mensagem retornada. Se a chamada falhar, as informações de grupo e segmento permanecerão as mesmas que antes da chamada.

### **MQGMO\_BROWSE\_NEXT**

Avance o cursor de navegação para a próxima mensagem na fila que satisfaz os critérios de seleção especificados na chamada MQGET. A mensagem é retornada ao aplicativo, mas permanece na fila.

MQGMO\_BROWSE\_NEXT não é válido com nenhuma das opções a seguir:

- MQGMO\_BROWSE\_FIRST
- MQGMO\_BROWSE\_MSG\_UNDER\_CURSOR
- MQGMO\_MARK\_SKIP\_BACKOUT
- MQGMO\_MSG\_UNDER\_CURSOR
- MQGMO\_SYNCPOINT
- MQGMO\_SYNCPOINT\_IF\_PERSISTENT
- MQGMO\_UNLOCK

Também será um erro se a fila não tiver sido aberta para procura.

MQGMO\_BROWSE\_NEXT se comporta da mesma maneira que MQGMO\_BROWSE\_FIRST, se for a primeira chamada para procurar uma fila, depois que a fila tiver sido aberta para procura.

A mensagem sob o cursor pode ser removida da fila antes da próxima chamada MQGET com MQGMO\_BROWSE\_NEXT ser emitida. O cursor de navegação permanece logicamente na posição na fila que a mensagem ocupou, mesmo que essa posição agora esteja vazia.

As mensagens são armazenadas na fila de uma das duas maneiras:

- FIFO dentro da prioridade (MQMDS\_PRIORITY) ou
- FIFO independentemente da prioridade (MQMDS\_FIFO)

O atributo da fila **MsgDeliverySequence** indica qual método se aplica (consulte [“Atributos para filas”](#) na página 865 para obter detalhes).

Uma fila pode ter um MsgDeliverySequence de MQMDS\_PRIORITY. Uma mensagem chega na fila que é de uma prioridade mais alta do que aquela apontada atualmente pelo cursor de navegação. Nesse caso, a mensagem de prioridade mais alta não será localizada durante a varredura atual da fila usando MQGMO\_BROWSE\_NEXT. Ele pode ser localizado somente após o cursor de navegação ter sido reconfigurado com MQGMO\_BROWSE\_FIRST ou reabrindo a fila..

A opção `MQGMO_MSG_UNDER_CURSOR` pode ser usada com uma chamada `MQGET` de não navegação, se necessário, para remover a mensagem da fila.

O cursor de navegação não é movido por chamadas `MQGET` não de navegação usando a mesma manipulação `Hobj`.

Especifique a opção `MQGMO_LOCK` com essa opção para bloquear a mensagem procurada.

É possível especificar `MQGMO_BROWSE_NEXT` com qualquer combinação válida das opções `MQGMO_*` e `MQMO_*` que controlam o processamento das mensagens em grupos e segmentos de mensagens lógicas.

Se você especificar `MQGMO_LOGICAL_ORDER`, as mensagens serão procuradas em ordem lógica. Se você omitir essa opção, as mensagens serão procuradas em ordem física. Se você especificar `MQGMO_BROWSE_FIRST`, será possível alternar entre ordem lógica e ordem física. Chamadas `MQGET` subsequentes usando `MQGMO_BROWSE_NEXT` navegam na fila na mesma ordem que a chamada mais recente que especificou `MQGMO_BROWSE_FIRST` para o identificador de filas. A chamada falha com o código de razão `MQRC_INCONSISTENT_BROWSE` se essa condição não for satisfeita.

**Nota:** Tome especial cuidado ao usar uma chamada `MQGET` para navegar além do término de um grupo de mensagens se `MQGMO_LOGICAL_ORDER` não for especificado. Por exemplo, suponha que a última mensagem no grupo precede a primeira mensagem no grupo na fila. Usando `MQGMO_BROWSE_NEXT` para navegar além do término do grupo, especificar `MQMO_MATCH_MSG_SEQ_NUMBER` com `MsgSeqNumber` configurado como 1 retorna a primeira mensagem no grupo já procurado. Esse resultado pode acontecer imediatamente, ou várias chamadas de `MQGET` posteriormente, se houver grupos intervenientes. A mesma consideração se aplica para uma mensagem lógica não em um grupo.

As informações de grupo e de segmento para chamadas de procura são retidas separadamente das informações para chamadas que removem mensagens da fila.

### **MQGMO\_BROWSE\_MSG\_UNDER\_CURSOR**

Recupere a mensagem apontada pelo cursor de procura não destrutivamente, independentemente das opções `MQMO_*` especificadas no campo `MatchOptions` em `MQGMO`.

`MQGMO_BROWSE_MSG_UNDER_CURSOR` não é válido com nenhuma das opções a seguir:

- `MQGMO_BROWSE_FIRST`
- `MQGMO_BROWSE_NEXT`
- `MQGMO_MARK_SKIP_BACKOUT`
- `MQGMO_MSG_UNDER_CURSOR`
- `MQGMO_SYNCPOINT`
- `MQGMO_SYNCPOINT_IF_PERSISTENT`
- `MQGMO_UNLOCK`

Também será um erro se a fila não tiver sido aberta para procura.

A mensagem apontada pelo cursor de navegação é aquela que foi recuperada pela última vez usando a opção `MQGMO_BROWSE_FIRST` ou `MQGMO_BROWSE_NEXT`. A chamada falhará se nenhuma dessas chamadas tiver sido emitida para essa fila desde que ela foi aberta. A chamada também falhará se a mensagem que estava sob o cursor de navegação tiver sido recuperada destrutivamente.

A posição do cursor de navegação não é alterada por esta chamada..

A opção `MQGMO_MSG_UNDER_CURSOR` pode ser usada com uma chamada `MQGET` não de procura, para remover a mensagem da fila.

O cursor de navegação não é movido por uma chamada `MQGET` não de navegação, mesmo se estiver usando a mesma manipulação `Hobj`. Também não é movido por uma chamada `MQGET` de navegação que retorna um código de conclusão de `MQCC_FAILED` ou um código de razão de `MQRC_TRUNCATED_MSG_FAILED`.

Se `MQGMO_BROWSE_MSG_UNDER_CURSOR` for especificado com `MQGMO_LOCK`:

- Se já houver uma mensagem bloqueada, ela deverá estar sob o cursor, para que seja retornada sem desbloqueio e bloqueio novamente. A mensagem permanece bloqueada..
- Se não houver mensagem bloqueada e houver uma mensagem sob o cursor de navegação, ela será bloqueada e retornada ao aplicativo. Se não houver nenhuma mensagem sob o cursor de navegação, a chamada falhará

Se MQGMO\_BROWSE\_MSG\_UNDER\_CURSOR for especificado sem MQGMO\_LOCK:

- Se já houver uma mensagem bloqueada, ela deverá ser aquela abaixo do cursor. A mensagem é retornada para o aplicativo e, em seguida, desbloqueada. Como a mensagem agora está desbloqueada, não há garantia de que ela possa ser procurada novamente ou recuperada destrutivamente pelo mesmo aplicativo. Ele pode ter sido recuperado destrutivamente por outro aplicativo obtendo mensagens da fila.
- Se não houver mensagem bloqueada e houver uma mensagem sob o cursor de navegação, ela será retornada ao aplicativo. Se não houver nenhuma mensagem sob o cursor de navegação, a chamada falhará

Se MQGMO\_COMPLETE\_MSG for especificado com MQGMO\_BROWSE\_MSG\_UNDER\_CURSOR, o cursor de navegação deverá identificar uma mensagem cujo campo Offset no MQMD seja zero. Se essa condição não for atendida, a chamada falhará com o código de razão MQRC\_INVALID\_MSG\_UNDER\_CURSOR

As informações de grupo e de segmento para chamadas de procura são retidas separadamente das informações para chamadas que removem mensagens da fila

### **MQGMO\_MSG\_UNDER\_CURSOR**

Recupere a mensagem apontada pelo cursor de navegação, independentemente das opções MQMO\_\* especificadas no campo MatchOptions em MQGMO. A mensagem é removida da fila.

A mensagem apontada pelo cursor de navegação é aquela que foi recuperada pela última vez usando a opção MQGMO\_BROWSE\_FIRST ou MQGMO\_BROWSE\_NEXT.

Se MQGMO\_COMPLETE\_MSG for especificado com MQGMO\_MSG\_UNDER\_CURSOR, o cursor de navegação deverá identificar uma mensagem cujo campo Offset no MQMD seja zero. Se essa condição não for atendida, a chamada falhará com o código de razão MQRC\_INVALID\_MSG\_UNDER\_CURSOR

Essa opção não é válida com nenhuma das opções a seguir:

- MQGMO\_BROWSE\_FIRST
- MQGMO\_BROWSE\_MSG\_UNDER\_CURSOR
- MQGMO\_BROWSE\_NEXT
- MQGMO\_UNLOCK

Também será um erro se a fila não tiver sido aberta para procura e para entrada. Se o cursor de navegação não estiver apontando atualmente para uma mensagem recuperável, um erro será retornado pela chamada MQGET.

### **MQGMO\_MARK\_BROWSE\_HANDLE**

A mensagem que é retornada por um MQGET bem-sucedido ou identificada pelo MsgToken retornado é marcada. A marca é específica para a manipulação de objetos usada na chamada.

A mensagem não é removida da fila.

MQGMO\_MARK\_BROWSE\_HANDLE será válido apenas se uma das seguintes opções também for especificada:

- MQGMO\_BROWSE\_FIRST
- MQGMO\_BROWSE\_MSG\_UNDER\_CURSOR
- MQGMO\_BROWSE\_NEXT

MQGMO\_MARK\_BROWSE\_HANDLE não é válido com nenhuma das opções a seguir:

- MQGMO\_ALL\_MSGS\_AVAILABLE

- MQGMO\_ALL\_SEGMENTS\_AVAILABLE
- MQGMO\_COMPLETE\_MSG
- MQGMO\_LOCK
- MQGMO\_LOGICAL\_ORDER
- MQGMO\_UNLOCK

A mensagem permanece nesse estado até que um dos seguintes eventos ocorra:

- A manipulação de objetos em questão é fechada, normalmente, ou de outra forma.
- A mensagem é desmarcada para essa manipulação por uma chamada para MQGET com a opção MQGMO\_UNMARK\_BROWSE\_HANDLE
- A mensagem é retornada de uma chamada para MQGET destrutivo, que é concluída com MQCC\_OK ou MQCC\_WARNING. O estado da mensagem permanece alterado mesmo se o MQGET for posteriormente recuperado.
- A mensagem expira..

### **MQGMO\_MARK\_BROWSE\_CO\_OP**

A mensagem que é retornada por um MQGET bem-sucedido ou identificada pelo *MsgToken* retornado é marcada para todas as manipulações no conjunto de cooperação.

A marca de nível cooperativo é além de qualquer marca de nível de manipulação que possa ter sido configurada.

A mensagem não é removida da fila

MQGMO\_MARK\_BROWSE\_CO\_OP será válido apenas se a manipulação de objetos usada tiver sido retornada por uma chamada para MQOPEN que especificou MQOO\_CO\_OP. Você também deve especificar uma das seguintes opções de MQGMO :

- MQGMO\_BROWSE\_FIRST
- MQGMO\_BROWSE\_MSG\_UNDER\_CURSOR
- MQGMO\_BROWSE\_NEXT

Essa opção não é válida com nenhuma das opções a seguir:

- MQGMO\_ALL\_MSGS\_AVAILABLE
- MQGMO\_ALL\_SEGMENTS\_AVAILABLE
- MQGMO\_COMPLETE\_MSG
- MQGMO\_LOCK
- MQGMO\_LOGICAL\_ORDER
- MQGMO\_UNLOCK

Se a mensagem já estiver marcada e a opção MQGMO\_UNMARKED\_BROWSE\_MSG não for especificada, a chamada falhará com MQCC\_FAILED e código de razão MQRC\_MSG\_MARKED\_BROWSE\_CO\_OP.

A mensagem permanece nesse estado até que um dos seguintes eventos ocorra:

- Todos os identificadores de objetos no conjunto de cooperação são fechados
- A mensagem é desmarcada para os navegadores cooperantes por uma chamada para MQGET com a opção MQGMO\_UNMARK\_BROWSE\_CO\_OP
- A mensagem é automaticamente desmarcada pelo gerenciador de filas...
- A mensagem é retornada de uma chamada para um MQGET que não seja de navegação. O estado da mensagem permanece alterado mesmo se o MQGET for posteriormente recuperado.
- A mensagem expira..

### **MQGMO\_UNMARKED\_BROWSE\_MSG**

Uma chamada para MQGET que especifica MQGMO\_UNMARKED\_BROWSE\_MSG retorna uma mensagem considerada desmarcada para sua manipulação. Ele não retorna uma mensagem se a mensagem foi

marcada para seu identificador. Ele também não retorna a mensagem se a fila foi aberta por uma chamada para MQOPEN, com a opção MQOO\_CO\_OP, e a mensagem foi marcada por um membro do conjunto de cooperação.

Essa opção não é válida com nenhuma das opções a seguir:

- MQGMO\_ALL\_MSGS\_AVAILABLE
- MQGMO\_ALL\_SEGMENTS\_AVAILABLE
- MQGMO\_COMPLETE\_MSG
- MQGMO\_LOCK
- MQGMO\_LOGICAL\_ORDER
- MQGMO\_UNLOCK

#### **MQGMO\_UNMARK\_BROWSE\_CO\_OP**

Após uma chamada para MQGET que especifica essa opção, a mensagem não é mais considerada por quaisquer identificadores abertos no conjunto de identificadores de cooperação a serem marcados para o conjunto de cooperação. A mensagem ainda será considerada marcada no nível do identificador se ela tiver sido marcada no nível do identificador antes desta chamada.

Usar MQGMO\_UNMARK\_BROWSE\_CO\_OP é válido apenas com um identificador retornado por uma chamada bem-sucedida para MQOPEN com a opção MQOO\_CO\_OP. O MQGET será bem-sucedido, mesmo se a mensagem não for considerada marcada pelo conjunto de identificadores cooperativo.

MQGMO\_UNMARK\_BROWSE\_CO\_OP não é válido em uma chamada MQGET não de procura ou com qualquer uma das opções a seguir:

- MQGMO\_ALL\_MSGS\_AVAILABLE
- MQGMO\_ALL\_SEGMENTS\_AVAILABLE
- MQGMO\_COMPLETE\_MSG
- MQGMO\_LOCK
- MQGMO\_LOGICAL\_ORDER
- MQGMO\_MARK\_BROWSE\_CO\_OP
- MQGMO\_UNLOCK
- MQGMO\_UNMARKED\_BROWSE\_MSG

#### **MQGMO\_UNMARK\_BROWSE\_HANDLE**

Após uma chamada para MQGET que especifica essa opção, a mensagem localizada não é mais considerada marcada por essa manipulação.

A chamada será bem-sucedida, mesmo se a mensagem não estiver marcada para essa manipulação.

Esta opção não é válida em uma chamada MQGET não de procura ou com qualquer uma das opções a seguir:

- MQGMO\_ALL\_MSGS\_AVAILABLE
- MQGMO\_ALL\_SEGMENTS\_AVAILABLE
- MQGMO\_COMPLETE\_MSG
- MQGMO\_LOCK
- MQGMO\_LOGICAL\_ORDER
- MQGMO\_MARK\_BROWSE\_CO\_OP
- MQGMO\_UNLOCK
- MQGMO\_UNMARKED\_BROWSE\_MSG

### **Opções de bloqueio**

As opções a seguir estão relacionadas a mensagens de bloqueio na fila:

## **MQGMO\_LOCK**

Bloqueie a mensagem que é procurada, para que a mensagem fique invisível para qualquer outro identificador aberto para a fila. A opção pode ser especificada apenas se uma das seguintes opções também for especificada:

- MQGMO\_BROWSE\_FIRST
- MQGMO\_BROWSE\_NEXT
- MQGMO\_BROWSE\_MSG\_UNDER\_CURSOR

Apenas uma mensagem pode ser bloqueada para cada identificador de fila. A mensagem pode ser uma mensagem lógica ou uma mensagem física:

- Se você especificar MQGMO\_COMPLETE\_MSG, todos os segmentos de mensagens que compõem a mensagem lógica serão bloqueados no identificador de filas. As mensagens devem estar presentes na fila e disponíveis para recuperação.
- Se você omitir MQGMO\_COMPLETE\_MSG, apenas uma única mensagem física será bloqueada para o identificador de filas. Se essa mensagem for um segmento de uma mensagem lógica, o segmento bloqueado evitará que outros aplicativos usem o MQGMO\_COMPLETE\_MSG para recuperar ou procurar a mensagem lógica.

A mensagem bloqueada é sempre aquela sob o cursor de navegação. A mensagem pode ser removida da fila por uma chamada MQGET posterior que especifica a opção MQGMO\_MSG\_UNDER\_CURSOR. Outras chamadas MQGET usando o identificador de fila também podem remover a mensagem (por exemplo, uma chamada que especifica o identificador de mensagem da mensagem bloqueada).

Se a chamada retornar o código de conclusão MQCC\_FAILED ou MQCC\_WARNING com código de razão MQRC\_TRUNCATED\_MSG\_FAILED, nenhuma mensagem será bloqueada.

Se o aplicativo não remover a mensagem da fila, o bloqueio será liberado por uma das seguintes ações:

- Emitir outra chamada MQGET para essa manipulação, especificando MQGMO\_BROWSE\_FIRST ou MQGMO\_BROWSE\_NEXT. O bloqueio será liberado se a chamada for concluída com MQCC\_OK ou MQCC\_WARNING. A mensagem permanece bloqueada se a chamada for concluída com MQCC\_FAILED. No entanto, as exceções a seguir se aplicam:
  - A mensagem não será desbloqueada se MQCC\_WARNING for retornado com MQRC\_TRUNCATED\_MSG\_FAILED.
  - A mensagem será desbloqueada se MQCC\_FAILED for retornado com MQRC\_NO\_MSG\_AVAILABLE.

Se você também especificar MQGMO\_LOCK, a mensagem retornada será bloqueada. Se você omitir MQGMO\_LOCK, não haverá nenhuma mensagem bloqueada após a chamada.

Se você especificar MQGMO\_WAIT e nenhuma mensagem estiver imediatamente disponível, a mensagem original será desbloqueada antes do início da espera.

- Emitir outra chamada MQGET para essa manipulação, com MQGMO\_BROWSE\_MSG\_UNDER\_CURSOR, sem MQGMO\_LOCK. O bloqueio será liberado se a chamada for concluída com MQCC\_OK ou MQCC\_WARNING. A mensagem permanece bloqueada se a chamada for concluída com MQCC\_FAILED. No entanto, a exceção a seguir se aplica:
  - A mensagem não será desbloqueada se MQCC\_WARNING for retornado com MQRC\_TRUNCATED\_MSG\_FAILED.
- Emitir outra chamada MQGET para essa manipulação com MQGMO\_UNLOCK.
- Emitir uma chamada MQCLOSE usando a manipulação. O MQCLOSE pode ser implícito, causado pelo término do aplicativo.

Nenhuma opção especial MQOPEN é necessária para especificar MQGMO\_LOCK, diferente de MQOO\_BROWSE, que é necessário para especificar uma opção de procura associada.

MQGMO\_LOCK não é válido com nenhuma das opções a seguir:

- MQGMO\_MARK\_SKIP\_BACKOUT
- MQGMO\_SYNCPOINT
- MQGMO\_SYNCPOINT\_IF\_PERSISTENT
- MQGMO\_UNLOCK

### **MQGMO\_UNLOCK**

A mensagem a ser desbloqueada deve ter sido bloqueada anteriormente por uma chamada de MQGET com a opção MQGMO\_LOCK. Se não houver nenhuma mensagem bloqueada para essa manipulação, a chamada será concluída com MQCC\_WARNING e MQRC\_NO\_MSG\_LOCKED.

Os parâmetros **MsgDesc**, **BufferLength**, **Buffer** e **DataLength** não serão verificados ou alterados se você especificar MQGMO\_UNLOCK.. Nenhuma mensagem é retornada em *Buffer*.

Nenhuma opção aberta especial é necessária para especificar MQGMO\_UNLOCK (embora MQOO\_BROWSE seja necessário para emitir a solicitação de bloqueio em primeiro lugar).

Essa opção não é válida com nenhuma opção, exceto a seguinte:

- MQGMO\_NO\_WAIT
- MQGMO\_NO\_SYNCPOINT

Ambas as opções são assumidas se especificadas ou não.

### **Opções de dados da mensagem.**

As opções a seguir estão relacionadas ao processamento dos dados da mensagem quando a mensagem é lida a partir da fila:

#### **MQGMO\_ACCEPT\_TRUNCATED\_MSG**

Se o buffer de mensagem for muito pequeno para conter a mensagem completa, permita que a chamada MQGET preencha o buffer.. MQGET preenche o buffer com o máximo de mensagens que ele pode. Ele emite um código de conclusão de aviso e conclui seu processamento.. Isto significa que:

- Ao navegar pelas mensagens, o cursor de navegação é avançado para a mensagem retornada
- Ao remover mensagens, a mensagem retornada é removida da fila.
- O código de razão MQRC\_TRUNCATED\_MSG\_ACCEPTED será retornado se nenhum outro erro ocorrer..

Sem essa opção, o buffer ainda é preenchido com o máximo de mensagens que ele pode conter.. Um código de conclusão de aviso é emitido, mas o processamento não é concluído. Isto significa que:

- Ao navegar pelas mensagens, o cursor de navegação não é avançado
- Ao remover mensagens, a mensagem não é removida da fila..
- O código de razão MQRC\_TRUNCATED\_MSG\_FAILED será retornado se não ocorrer nenhum outro erro

#### **MQGMO\_CONVERT**

Essa opção converte os dados do aplicativo na mensagem para conformidade com os valores CodedCharSetId e Encoding especificados no parâmetro **MsgDesc** na chamada MQGET . Os dados são convertidos antes de serem copiados para o parâmetro **Buffer**.

O campo Format especificado quando a mensagem foi colocada é assumido pelo processo de conversão para identificar a natureza dos dados na mensagem. Os dados da mensagem são convertidos pelo gerenciador de filas para formatos integrados e por uma saída gravada pelo usuário para outros formatos.. Consulte [“saída de conversão de dados” na página 939](#) para obter detalhes da saída de conversão de dados.

- Se a conversão for bem-sucedida, os campos CodedCharSetId e Encoding especificados no parâmetro **MsgDesc** serão inalterados no retorno da chamada MQGET .

- Se apenas a conversão falhar, os dados da mensagem serão retornados sem conversão. Os campos `CodedCharSetId` e `Encoding` em `MsgDesc` serão configurados para os valores para a mensagem não convertida. Neste caso, o código de conclusão é `MQCC_WARNING`.

Em qualquer caso, esses campos descrevem o identificador do conjunto de caracteres e a codificação dos dados da mensagem que são retornados no parâmetro **Buffer**.

Consulte o campo *Format* descrito em “MQMD - descritor de mensagem” na página 435 para obter uma lista de nomes de formato para os quais o gerenciador de fila executa a conversão.

## Opções de grupo e segmento

As opções a seguir estão relacionadas ao processamento de mensagens em grupos e segmentos de mensagens lógicas. Antes das descrições de opções, aqui estão algumas definições de termos importantes:

### Mensagem física

Uma mensagem física é a menor unidade de informações que pode ser colocada em ou removida de uma fila. Geralmente, ele corresponde às informações especificadas ou recuperadas em uma única chamada `MQPUT`, `MQPUT1` ou `MQGET`. Cada mensagem física possui seu descritor de mensagens, `MQMD`. Geralmente, as mensagens físicas são distinguidas por valores diferentes para o identificador de mensagem, o campo `MsgId` em `MQMD`. O gerenciador de filas não aplica valores diferentes.

### Mensagem lógica

Uma mensagem lógica é uma única unidade de informações do aplicativo. Na ausência de restrições do sistema, uma mensagem lógica é a mesma que uma mensagem física. Se as mensagens lógicas forem grandes, as restrições do sistema poderão tornar aconselhável ou necessário dividir uma mensagem lógica em duas ou mais mensagens físicas, chamadas segmentos.

Uma mensagem lógica que foi segmentada consiste em duas ou mais mensagens físicas que possuem o mesmo identificador de grupo não nulo, campo `GroupId` em `MQMD`. Eles possuem o mesmo número de sequência da mensagem, campo `MsgSeqNumber` no `MQMD`. Os segmentos são diferenciados por valores diferentes para o deslocamento do segmento, campo `Offset` no `MQMD`. O deslocamento de segmento é o deslocamento dos dados na mensagem física do início dos dados na mensagem lógica. Como cada segmento é uma mensagem física, os segmentos em uma mensagem lógica geralmente possuem identificadores de mensagem diferentes.

Uma mensagem lógica que não foi segmentada, mas cuja segmentação foi permitida pelo aplicativo de envio, também possui um identificador de grupo não nulo. Nesse caso, há apenas uma mensagem física com esse identificador de grupo se a mensagem lógica não pertencer a um grupo de mensagens. Mensagens lógicas, para as quais a segmentação foi inibida pelo aplicativo de envio, possuem um identificador de grupo nulo, `MQGI_NONE`, a menos que a mensagem lógica pertença a um grupo de mensagens.

### Grupo de mensagens

Um grupo de mensagens é um conjunto de uma ou mais mensagens lógicas que possuem o mesmo identificador de grupo não nulo. As mensagens lógicas no grupo são distinguidas por valores diferentes para o número de sequência da mensagem. O número de sequência é um inteiro no intervalo de 1 a *n*, em que *n* é o número de mensagens lógicas no grupo. Se uma ou mais mensagens lógicas forem segmentadas, haverá mais de *n* mensagens físicas no grupo.

### MQGMO\_LOGICAL\_ORDER

`MQGMO_LOGICAL_ORDER` controla a ordem na qual as mensagens são retornadas por chamadas `MQGET` sucessivas para o identificador de fila. A opção deve ser especificada em cada chamada.

Se `MQGMO_LOGICAL_ORDER` for especificado para chamadas `MQGET` sucessivas para o mesmo identificador de filas, as mensagens em grupos serão retornadas na ordem de seus números de sequência de mensagem. Segmentos de mensagens lógicas são retornados na ordem fornecida por seus deslocamentos de segmentos. Essa ordem pode ser diferente da ordem na qual essas mensagens e segmentos ocorrem na fila.

**Nota:** Especificar `MQGMO_LOGICAL_ORDER` não tem consequências adversas em mensagens que não pertencem a grupos e que não são segmentos. Com efeito, essas mensagens são tratadas como se



cada uma pertencesse a um grupo de mensagens que consiste em apenas uma mensagem. É seguro especificar MQGMO\_LOGICAL\_ORDER ao recuperar mensagens de filas que contêm uma combinação de mensagens em grupos, segmentos de mensagens e mensagens não segmentadas que não estão em grupos.

Para retornar as mensagens na ordem necessária, o gerenciador de filas retém as informações de grupo e segmento entre sucessivas chamadas MQGET. As informações de grupo e segmento identificam o grupo de mensagens atual e a mensagem lógica atual para o identificador de filas. Ele também identifica a posição atual dentro do grupo e da mensagem lógica e se as mensagens estão sendo recuperadas dentro de uma unidade de trabalho. Como o gerenciador de filas retém essas informações, o aplicativo não precisa configurar as informações de grupo e segmento antes de cada chamada MQGET. Especificamente, isso significa que o aplicativo não precisa configurar os campos GroupId, MsgSeqNumber e Offset no MQMD. No entanto, o aplicativo deve configurar a opção MQGMO\_SYNCPOINT ou MQGMO\_NO\_SYNCPOINT corretamente em cada chamada.

Quando a fila é aberta não há grupo de mensagens atual e nenhuma mensagem lógica atual. Um grupo de mensagens se torna o grupo de mensagens atual quando uma mensagem que possui a sinalização MQMF\_MSG\_IN\_GROUP é retornada pela chamada MQGET. Com MQGMO\_LOGICAL\_ORDER especificado em chamadas sucessivas, esse grupo permanece o grupo atual até que seja retornada uma mensagem que tenha:

- MQMF\_LAST\_MSG\_IN\_GROUP sem MQMF\_SEGMENT (ou seja, a última mensagem lógica no grupo não é segmentada) ou
- MQMF\_LAST\_MSG\_IN\_GROUP com MQMF\_LAST\_SEGMENT (ou seja, a mensagem retornada é o último segmento da última mensagem lógica no grupo).

Quando essa mensagem é retornada, o grupo de mensagens é finalizado e, na conclusão bem-sucedida da chamada MQGET, não há mais um grupo atual. De maneira semelhante, uma mensagem lógica torna-se a mensagem lógica atual quando uma mensagem que possui o sinalizador MQMF\_SEGMENT é retornada pela chamada MQGET. A mensagem lógica é finalizada quando a mensagem que possui o sinalizador MQMF\_LAST\_SEGMENT é retornada.

Se nenhum critério de seleção for especificado, chamadas MQGET sucessivas retornarão, na ordem correta, as mensagens para o primeiro grupo de mensagens na fila. Em seguida, eles retornam as mensagens para o segundo grupo de mensagens, e assim por diante, até que não haja mais mensagens disponíveis. É possível selecionar os grupos de mensagens específicos retornados especificando uma ou mais das seguintes opções no campo MatchOptions:

- MQMO\_MATCH\_MSG\_ID
- MQMO\_MATCH\_CORREL\_ID
- MQMO\_MATCH\_GROUP\_ID

No entanto, essas opções são efetivas apenas quando não há nenhum grupo de mensagens ou mensagem lógica atual. Consulte o campo MatchOptions descrito em [“MQGMO-Obter opções de mensagem” na página 379](#), para obter detalhes adicionais.

Tabela 495 na página 406 mostra os valores dos campos MsgId, CorrelId, GroupId, MsgSeqNumber e Offset que o gerenciador de filas procura ao tentar localizar uma mensagem para retornar na chamada MQGET. As regras se aplicam para remover mensagens da fila e procurar mensagens na fila. Na tabela, significa Sim ou Não:

#### **LOG ORD**

Indica se a opção MQGMO\_LOGICAL\_ORDER é especificada na chamada.

#### **Cur grp**

Indica se um grupo de mensagens atual existe antes da chamada.

#### **Cur log msg**

Indica se uma mensagem lógica atual existe antes da chamada.

#### **Outras colunas**

Mostrar os valores que o Gerenciador de Filas procura. Anterior indica o valor retornado para o campo na mensagem anterior para o identificador de filas.

Tabela 495. Opções MQGET relacionadas a mensagens em grupos e segmentos de mensagens lógicas

Opções que você especifica	Grupo e status log-msg antes da chamada		Valores que o gerenciador de filas procura				
	LOG ORD	Cur grp	Cur log msg	MsgId	CorrelId	GroupId	MsgSeqNumber
Sim	No	No	Controlado por <i>MatchOptions</i>	Controlado por <i>MatchOptions</i>	Controlado por <i>MatchOptions</i>	1	0
Sim	No	Sim	Qualquer identificador de mensagem	Qualquer identificador de correlação	Identificador do grupo anterior	1	Deslocamento anterior + comprimento do segmento anterior
Sim	Sim	No	Qualquer identificador de mensagem	Qualquer identificador de correlação	Identificador do grupo anterior	Número da sequência anterior + 1	0
Sim	Sim	Sim	Qualquer identificador de mensagem	Qualquer identificador de correlação	Identificador do grupo anterior	Número de sequência anterior	Deslocamento anterior + comprimento do segmento anterior
No	Qualquer um	Qualquer um	Controlado por <i>MatchOptions</i>	Controlado por <i>MatchOptions</i>	Controlado por <i>MatchOptions</i>	Controlado por <i>MatchOptions</i>	Controlado por <i>MatchOptions</i>

Se vários grupos de mensagens estiverem presentes na fila e elegíveis para retorno, os grupos serão retornados na ordem determinada pela posição na fila do primeiro segmento da primeira mensagem lógica em cada grupo. Ou seja, as mensagens físicas que possuem números de sequência de mensagens de 1 e deslocamentos de 0, determinam a ordem na qual os grupos elegíveis são retornados.

A opção MQGMO\_LOGICAL\_ORDER afeta unidades de trabalho da seguinte forma:

- Se a primeira mensagem lógica ou segmento em um grupo for recuperado em uma unidade de trabalho, todas as outras mensagens lógicas e segmentos no grupo deverão ser recuperados em uma unidade de trabalho, se o mesmo identificador de fila for usado. No entanto, eles não precisam ser recuperados dentro da mesma unidade de trabalho. Isso permite que um grupo de mensagens que consiste em muitas mensagens físicas seja dividido em duas ou mais unidades consecutivas de trabalho para o identificador da fila.
- Se a primeira mensagem lógica ou segmento em um grupo não for recuperado em uma unidade de trabalho e a mesma manipulação de filas for utilizada, nenhuma das outras mensagens lógicas e segmentos no grupo poderão ser recuperados em uma unidade de trabalho.

Se essas condições não forem satisfeitas, a chamada MQGET falha com o código de razão MQRC\_INCONSISTENT\_UOW.

Quando MQGMO\_LOGICAL\_ORDER é especificado, o MQGMO fornecido na chamada MQGET não deve ser menor que MQGMO\_VERSION\_2e o MQMD não deve ser menor que MQMD\_VERSION\_2.. Se essa condição não for atendida, a chamada falhará com o código de razão MQRC\_WRONG\_GMO\_VERSION ou MQRC\_WRONG\_MD\_VERSION, conforme apropriado.

Se MQGMO\_LOGICAL\_ORDER não for especificado para chamadas MQGET sucessivas para o identificador de filas, as mensagens serão retornadas sem considerar se pertencem a grupos de mensagens ou se são segmentos de mensagens lógicas. Isso significa que mensagens ou segmentos de um determinado grupo ou mensagem lógica podem ser retornados fora de ordem ou misturados com mensagens ou segmentos de outros grupos ou mensagens lógicas ou com mensagens que não estão em grupos e não são segmentos. Nessa situação, as mensagens específicas que são retornadas pelas sucessivas chamadas MQGET são controladas pelas opções MQMO\_\* especificadas nessas chamadas (consulte o campo *MatchOptions* descrito em “MQGMO-Obter opções de mensagem” na página 379 para obter detalhes dessas opções).

Esta é a técnica que pode ser utilizada para reiniciar um grupo de mensagens ou mensagem lógica no meio, após uma falha do sistema ter ocorrido. Quando o sistema reiniciar, o aplicativo poderá configurar os campos GroupId, MsgSeqNumber, Offset e MatchOptions para os valores apropriados e, em seguida, emitir a chamada MQGET com MQGMO\_SYNCPOINT ou MQGMO\_NO\_SYNCPOINT configurado, mas sem especificar MQGMO\_LOGICAL\_ORDER. Se essa chamada for bem-sucedida, o gerenciador de filas reterá as informações do grupo e do segmento e as chamadas subsequentes do MQGET que usam essa manipulação de filas poderão especificar MQGMO\_LOGICAL\_ORDER como normais

As informações do grupo e do segmento que o gerenciador de filas retém para a chamada MQGET são separadas das informações do grupo e do segmento que ele retém para a chamada MQPUT Além disso, o gerenciador de filas retém informações separadas para:

- Chamadas de MQGET que removem mensagens da fila
- MQGET chamadas que procuram mensagens na fila.

Para qualquer identificador de fila especificado, o aplicativo pode combinar chamadas MQGET que especificam MQGMO\_LOGICAL\_ORDER com chamadas MQGET que não. No entanto, observe os seguintes pontos:

- Se você omitir MQGMO\_LOGICAL\_ORDER, cada chamada MQGET bem-sucedida fará com que o gerenciador de fila configure as informações salvas do grupo e do segmento para os valores correspondentes à mensagem retornada; isso substitui as informações existentes do grupo e do segmento retidas pelo gerenciador de filas para a manipulação de filas. Apenas as informações apropriadas para a ação da chamada (procurar ou remover) são modificadas.
- Se você omitir MQGMO\_LOGICAL\_ORDER, a chamada não falhará se houver um grupo de mensagens ou uma mensagem lógica atual; a chamada poderá ser bem-sucedida com um código de conclusão MQCC\_WARNING O Tabela 496 na página 407 mostra os diferentes casos que podem surgir. Nesses casos, se o código de conclusão não for MQCC\_OK, o código de razão será um dos seguintes (conforme apropriado):
  - MQRC\_INCOMPLETE\_GROUP
  - MQRC\_INCOMPLETE\_MSG
  - MQRC\_INCONSISTENT\_UOW

**Nota:** O gerenciador de filas não verifica as informações do grupo e do segmento ao navegar em uma fila ou ao fechar uma fila que foi aberta para navegação, mas não entrada; nesses casos, o código de conclusão é sempre MQCC\_OK (assumindo nenhum outro erro).

Tabela 496. Resultado quando a chamada MQGET ou MQCLOSE não é consistente com as informações de grupo e segmento

A chamada atual é	A chamada anterior era MQGET com MQGMO_LOGICAL_ORDER	A chamada anterior era MQGET sem MQGMO_LOGICAL_ORDER
MQGET com MQGMO_LOGICAL_ORDER	MQCC_FAILED	MQCC_FAILED
MQGET sem MQGMO_LOGICAL_ORDER	MQCC_WARNING	MQCC_OK


Tabela 496. Resultado quando a chamada MQGET ou MQCLOSE não é consistente com as informações de grupo e segmento (continuação)

A chamada atual é	A chamada anterior era MQGET com MQGMO_LOGICAL_ORDER	A chamada anterior era MQGET sem MQGMO_LOGICAL_ORDER
MQCLOSE com um grupo ou mensagem lógica não terminada	MQCC_WARNING	MQCC_OK

Os aplicativos que desejam recuperar mensagens e segmentos em ordem lógica são recomendados para especificar MQGMO\_LOGICAL\_ORDER, já que essa é a opção mais simples de usar. Esta opção livra o aplicativo da necessidade de gerenciar as informações de grupo e segmento, pois o gerenciador de filas gerencia essa informação. No entanto, os aplicativos especializados podem precisar de mais controle do que o fornecido pela opção MQGMO\_LOGICAL\_ORDER, e isso pode ser obtido não especificando essa opção. O aplicativo deve então assegurar que os campos MsgId, CorrelId, GroupId, MsgSeqNumber e Offset em MQMD de as opções MQMO\_\* em MatchOptions em MQGMO sejam configurados corretamente, antes de cada chamada MQGET.

Por exemplo, um aplicativo que deseja encaminhar mensagens físicas que recebe, sem considerar se essas mensagens estão em grupos ou segmentos de mensagens lógicas, não deve especificar MQGMO\_LOGICAL\_ORDER. Em uma rede complexa com vários caminhos entre os gerenciadores de filas de envio e recebimento, as mensagens físicas podem chegar fora de ordem. Ao especificar nem MQGMO\_LOGICAL\_ORDER, nem o MQPMO\_LOGICAL\_ORDER correspondente na chamada MQPUT, o aplicativo de encaminhamento pode recuperar e encaminhar cada mensagem física assim que ela chegar, sem ter que aguardar a próxima na ordem lógica de chegada.

É possível especificar MQGMO\_LOGICAL\_ORDER com qualquer uma das outras opções MQGMO\_\* e com várias das opções MQMO\_\* em circunstâncias apropriadas (consulte a seção anterior).

-  No z/OS, essa opção é suportada para filas privadas e compartilhadas, mas a fila deve ter um tipo de índice de MQIT\_GROUP\_ID. Para filas compartilhadas, o objeto CFSTRUCT para o qual a fila é mapeada deve estar em CFLEVEL (3) ou superior.

- Essa opção é suportada para todas as filas locais para as seguintes plataformas:

-  AIX
-  Linux
-  IBM i
-  Windows

e para o IBM MQ MQI clients conectado a esses sistemas,

### MQGMO\_COMPLETE\_MSG

Apenas uma mensagem lógica completa pode ser retornada pela chamada MQGET. Se a mensagem lógica for segmentada, o gerenciador de filas remonta os segmentos e retorna a mensagem lógica completa para o aplicativo; o fato de que a mensagem lógica foi segmentada não é aparente para o aplicativo que a recupera.

**Nota:** Esta é a única opção que faz com que o gerenciador de filas remonte segmentos de mensagens. Se não for especificado, os segmentos serão retornados individualmente para o aplicativo se eles estiverem presentes na fila (e eles satisfizerem os outros critérios de seleção especificados na chamada MQGET). Aplicativos que não desejam receber segmentos individuais devem sempre especificar MQGMO\_COMPLETE\_MSG.

Para usar essa opção, o aplicativo deve fornecer um buffer grande o suficiente para acomodar a mensagem completa ou especificar a opção MQGMO\_ACCEPT\_TRUNCATED\_MSG.

Se a fila contiver mensagens segmentadas com alguns dos segmentos ausentes (talvez porque eles foram atrasados na rede e ainda não chegaram), especificar MQGMO\_COMPLETE\_MSG evitará a recuperação de segmentos pertencentes a mensagens lógicas incompletas. No entanto, esses

segmentos de mensagem ainda contribuem para o valor do atributo da fila **CurrentQDepth** ; isso significa que pode não haver mensagens lógicas recuperáveis, embora *CurrentQDepth* seja maior que zero.

Para mensagens persistentes, o gerenciador de filas pode remontar os segmentos apenas dentro de uma unidade de trabalho:

- Se a chamada MQGET estiver operando em uma unidade definida pelo usuário, essa unidade de trabalho será usada. Se a chamada falhar durante o processo de remontagem, o gerenciador de filas restabelece na fila quaisquer segmentos que foram removidos durante a remontagem. Entretanto, a falha não impede que a unidade de trabalho seja confirmada com êxito.
- Se a chamada estiver operando fora de uma unidade de trabalho definida pelo usuário e não existir nenhuma unidade de trabalho definida pelo usuário, o gerenciador de filas criará uma unidade de trabalho para a duração da chamada. Se a chamada for bem-sucedida, o gerenciador de fila confirma a unidade de trabalho automaticamente (o aplicativo não precisa fazer isso). Se a chamada falhar, o gerenciador de filas volta a unidade de trabalho.
- Se a chamada estiver operando fora de uma unidade de trabalho definida pelo usuário, mas uma unidade de trabalho definida pelo usuário existir, o gerenciador de filas não poderá remontar. Se a mensagem não precisar de remontagem, a chamada ainda poderá ser bem-sucedida Mas se a mensagem requer remontagem, a chamada falha com o código de razão MQRC\_UOW\_NOT\_AVAILABLE.

Para mensagens não persistentes, o gerenciador de filas não requer que uma unidade de trabalho esteja disponível para executar a remontagem.

Cada mensagem física que é um segmento possui seu próprio descritor de mensagens Para os segmentos que constituem uma mensagem lógica única, a maioria dos campos no descritor de mensagens são os mesmos para todos os segmentos na mensagem lógica; geralmente são apenas os campos `MsgId`, `Offset` e `MsgFlags` que diferem entre os segmentos na mensagem lógica. No entanto, se um segmento for colocado em uma fila de devoluções em um gerenciador de filas intermediário, o manipulador de fila de mensagens não entregues recuperará a mensagem especificando a opção `MQGMO_CONVERT` e isso poderá resultar no conjunto de caracteres ou na codificação do segmento sendo alterado. Se o manipulador de fila de mensagens não entregues enviar com êxito o segmento em seu caminho, o segmento poderá ter um conjunto de caracteres ou uma codificação diferente dos outros segmentos na mensagem lógica quando o segmento chegar ao gerenciador de filas de destino.

Uma mensagem lógica consistindo em segmentos nos quais os campos `CodedCharSetId` e `Encoding` diferem não pode ser remontada pelo gerenciador de fila em uma única mensagem lógica.. Em vez disso, o gerenciador de filas remonta e retorna os primeiros segmentos consecutivos no início da mensagem lógica que possuem os mesmos identificadores e codificações de conjunto de caracteres e a chamada MQGET é concluída com o código de conclusão `MQCC_WARNING` e o código de razão `MQRC_INCONSISTENT_CCIDS` ou `MQRC_INCONSISTENT_ENCODINGS`, conforme apropriado. Isso acontece independentemente de o `MQGMO_CONVERT` ser especificado. Para recuperar os segmentos restantes, o aplicativo deve emitir novamente a chamada MQGET sem a opção `MQGMO_COMPLETE_MSG` , recuperando os segmentos um por um.. O `MQGMO_LOGICAL_ORDER` pode ser usado para recuperar os segmentos restantes em ordem.

Um aplicativo que coloca segmentos também pode configurar outros campos no descritor de mensagem para valores que diferem entre segmentos. Entretanto, não há vantagem em fazer isso se o aplicativo de recebimento usar `MQGMO_COMPLETE_MSG` para recuperar a mensagem lógica. Quando o gerenciador de filas remonta uma mensagem lógica, ele retorna no descritor de mensagens os valores do descritor de mensagem para o primeiro segmento; a única exceção é o campo `MsgFlags` , que o gerenciador de filas configura para indicar que a mensagem remontada é o único segmento.


Se `MQGMO_COMPLETE_MSG` for especificado para uma mensagem de relatório, o gerenciador de filas executará um processamento especial O gerenciador de filas verifica a fila para ver se todas as mensagens de relatório desse tipo de relatório relacionadas aos diferentes segmentos na mensagem lógica estão presentes na fila.. Se estiverem, eles poderão ser recuperados como uma única mensagem, especificando `MQGMO_COMPLETE_MSG` Para que isso seja possível, as mensagens de

relatório devem ser geradas por um gerenciador de filas ou MCA que suporta segmentação ou o aplicativo de origem deve solicitar pelo menos 100 bytes de dados da mensagem (ou seja, as opções MQRO\_\*\_WITH\_DATA ou MQRO\_\*\_WITH\_FULL\_DATA apropriadas devem ser especificadas). Se a quantidade total de dados do aplicativo estiver presente para um segmento, os bytes ausentes serão substituídos por nulos na mensagem de relatório retornada.

Se MQGMO\_COMPLETE\_MSG for especificado com MQGMO\_MSG\_UNDER\_CURSOR ou MQGMO\_BROWSE\_MSG\_UNDER\_CURSOR, o cursor de navegação deverá ser posicionado em uma mensagem cujo campo *Offset* no MQMD tenha um valor 0. Se essa condição não for atendida, a chamada falhará com o código de razão MQRC\_INVALID\_MSG\_UNDER\_CURSOR

MQGMO\_COMPLETE\_MSG implica MQGMO\_ALL\_SEGMENTS\_AVAILABLE, que portanto não precisa ser especificado.

MQGMO\_COMPLETE\_MSG pode ser especificado com qualquer uma das outras opções MQGMO\_\* além de MQGMO\_SYNCPOINT\_IF\_PERSISTENT com qualquer uma das opções MQMO\_\* além de MQMO\_MATCH\_OFFSET.

-  No z/OS, essa opção é suportada para filas privadas e compartilhadas, mas a fila deve ter um tipo de índice MQIT\_GROUP\_ID. Para filas compartilhadas, o objeto CFSTRUCT para o qual o mapa de fila deve estar em CFLEVEL (3) ou superior...

- Nas seguintes plataformas:

-  AIX
-  IBM i
-  Linux
-  Windows

e para o IBM MQ MQI clients conectado a esses sistemas, essa opção é suportada para todas as filas locais

### MQGMO\_ALL\_MSGS\_AVAILABLE

As mensagens em um grupo tornam-se disponíveis para recuperação apenas quando todas as mensagens no grupo estão disponíveis. Se a fila contiver grupos de mensagens com algumas das mensagens ausentes (talvez porque elas foram atrasadas na rede e ainda não chegaram), especificar MQGMO\_ALL\_MSGS\_AVAILABLE evitará a recuperação de mensagens pertencentes a grupos incompletos. No entanto, essas mensagens ainda contribuem para o valor do atributo da fila **CurrentQDepth**; isso significa que pode não haver grupos de mensagens recuperáveis, mesmo que CurrentQDepth seja maior que zero. Se não houver nenhuma outra mensagem recuperável, o código de razão MQRC\_NO\_MSG\_AVAILABLE será retornado após o intervalo de espera especificado (se houver) ter expirado.

O processamento de MQGMO\_ALL\_MSGS\_AVAILABLE depende se MQGMO\_LOGICAL\_ORDER também é especificado:


- Se ambas as opções forem especificadas, MQGMO\_ALL\_MSGS\_AVAILABLE terá um efeito apenas quando não houver nenhum grupo atual ou mensagem lógica. Se houver um grupo atual ou mensagem lógica, MQGMO\_ALL\_MSGS\_AVAILABLE será ignorado. Isso significa que MQGMO\_ALL\_MSGS\_AVAILABLE pode permanecer ao processar mensagens em ordem lógica.
- Se MQGMO\_ALL\_MSGS\_AVAILABLE for especificado sem MQGMO\_LOGICAL\_ORDER, MQGMO\_ALL\_MSGS\_AVAILABLE sempre terá efeito. Isso significa que a opção deve ser desativada após a primeira mensagem no grupo ter sido removida da fila, para poder remover as mensagens restantes no grupo.

A conclusão bem-sucedida de uma chamada MQGET especificando MQGMO\_ALL\_MSGS\_AVAILABLE significa que no momento em que a chamada MQGET foi emitida, todas as mensagens no grupo estavam na fila. No entanto, esteja ciente de que outros aplicativos ainda podem remover mensagens do grupo (o grupo não está bloqueado para o aplicativo que recupera a primeira mensagem no grupo).

Se você omitir essa opção, as mensagens pertencentes a grupos poderão ser recuperadas mesmo quando o grupo estiver incompleto

MQGMO\_ALL\_MSGS\_AVAILABLE implica MQGMO\_ALL\_SEGMENTS\_AVAILABLE, que portanto não precisa ser especificado.

MQGMO\_ALL\_MSGS\_AVAILABLE pode ser especificado com qualquer uma das outras opções MQGMO\_\* e com qualquer uma das opções MQMO\_\* .

-  No z/OS, essa opção é suportada para filas privadas e compartilhadas, mas a fila deve ter um tipo de índice MQIT\_GROUP\_ID. Para filas compartilhadas, o objeto CFSTRUCT para o qual o mapa de fila deve estar em CFLEVEL (3) ou superior...

• Nas seguintes plataformas:

-  AIX
-  IBM i
-  Linux
-  Windows

e para o IBM MQ MQI clients conectado a esses sistemas, essa opção é suportada para todas as filas locais

### MQGMO\_ALL\_SEGMENTS\_AVAILABLE

Segmentos em uma mensagem lógica se tornam disponíveis para recuperação apenas quando todos os segmentos na mensagem lógica estão disponíveis. Se a fila contiver mensagens segmentadas com alguns dos segmentos ausentes (talvez porque eles foram atrasados na rede e ainda não chegaram), especificar MQGMO\_ALL\_SEGMENTS\_AVAILABLE evitará a recuperação de segmentos pertencentes a mensagens lógicas incompletas. No entanto, esses segmentos ainda contribuem para o valor do atributo da fila **CurrentQDepth** ; isso significa que pode não haver mensagens lógicas recuperáveis, embora CurrentQDepth seja maior que zero. Se não houver nenhuma outra mensagem recuperável, o código de razão MQRC\_NO\_MSG\_AVAILABLE será retornado após o intervalo de espera especificado (se houver) ter expirado.

O processamento de MQGMO\_ALL\_SEGMENTS\_AVAILABLE depende se MQGMO\_LOGICAL\_ORDER também é especificado:

- Se ambas as opções forem especificadas, MQGMO\_ALL\_SEGMENTS\_AVAILABLE terá efeito apenas quando não houver nenhuma mensagem lógica atual. Se houver uma mensagem lógica atual, MQGMO\_ALL\_SEGMENTS\_AVAILABLE será ignorado.. Isso significa que MQGMO\_ALL\_SEGMENTS\_AVAILABLE pode permanecer ao processar mensagens em ordem lógica.
- Se MQGMO\_ALL\_SEGMENTS\_AVAILABLE for especificado sem MQGMO\_LOGICAL\_ORDER, MQGMO\_ALL\_SEGMENTS\_AVAILABLE sempre terá efeito. Isso significa que a opção deve ser desativada após o primeiro segmento na mensagem lógica ter sido removido da fila, para poder remover os segmentos restantes na mensagem lógica.

Se esta opção não for especificada, os segmentos de mensagem poderão ser recuperados mesmo quando a mensagem lógica estiver incompleta...

Embora o MQGMO\_COMPLETE\_MSG e o MQGMO\_ALL\_SEGMENTS\_AVAILABLE requeiram que todos os segmentos estejam disponíveis antes que qualquer um deles possa ser recuperado, o primeiro retorna a mensagem completa, enquanto o último permite que os segmentos sejam recuperados um por um.

Se MQGMO\_ALL\_SEGMENTS\_AVAILABLE for especificado para uma mensagem de relatório, o gerenciador de filas verifica a fila para ver se há pelo menos uma mensagem de relatório para cada um dos segmentos que compõem a mensagem lógica completa. Se houver, a condição MQGMO\_ALL\_SEGMENTS\_AVAILABLE será atendida. No entanto, o gerenciador de filas não verifica o *tipo* das mensagens de relatório presentes e, portanto, pode haver uma combinação de tipos de relatório nas mensagens de relatório relacionadas aos segmentos da mensagem



lógica. Como resultado, o sucesso do MQGMO\_ALL\_SEGMENTS\_AVAILABLE não implica que o MQGMO\_COMPLETE\_MSG será bem-sucedido. Se houver uma combinação de tipos de relatórios presentes para os segmentos de uma mensagem lógica específica, essas mensagens de relatórios devem ser recuperados uma por uma.

É possível especificar MQGMO\_ALL\_SEGMENTS\_AVAILABLE com qualquer uma das outras opções MQGMO\_\* e com qualquer uma das opções MQMO\_\*.

- No z/OS, essa opção é suportada para filas privadas e compartilhadas, mas a fila deve ter um tipo de índice MQIT\_GROUP\_ID. Para filas compartilhadas, o objeto CFSTRUCT para o qual o mapa de fila deve estar em CFLEVEL (3) ou superior...
- Nas seguintes plataformas:

-  AIX
-  IBM i
-  Linux
-  Windows

e para o IBM MQ MQI clients conectado a esses sistemas, essa opção é suportada para todas as filas locais

## Opções de propriedade

As opções a seguir são relacionadas às propriedades da mensagem:

### MQGMO\_PROPERTIES\_AS\_Q\_DEF

Propriedades da mensagem, exceto aquelas contidas no descritor de mensagens (ou extensão) devem ser representadas conforme definido pelo atributo da fila **PropertyControl**. Se `MsgHandle` for fornecido, essa opção será ignorada e as propriedades da mensagem estarão disponíveis por meio do `MsgHandle`, a menos que o valor do atributo da fila **PropertyControl** seja `MQPROP_FORCE_MQRFH2`.

Essa será a ação padrão, se nenhuma opção da propriedade estiver especificada.

### MQGMO\_PROPERTIES\_IN\_HANDLE

As propriedades da mensagem devem ser disponibilizadas por meio do `MsgHandle`. Se não for fornecido nenhum identificador de mensagem, a chamada falhará com a razão `MQRC_HMSG_ERROR`.

**Nota:** Se a mensagem for lida posteriormente por um aplicativo que não cria um identificador de mensagens, o gerenciador de fila colocará quaisquer propriedades de mensagem em uma estrutura `MQRFH2`. Você pode achar que a presença de um cabeçalho `MQRFH2` inesperado interrompe o comportamento de um aplicativo existente.

### MQGMO\_NO\_PROPERTIES

Nenhuma propriedade da mensagem, exceto aquelas contidas no descritor de mensagem (ou extensão), será recuperada. Se um `MsgHandle` for fornecido ele será ignorado.

### MQGMO\_PROPERTIES\_FORCE\_MQRFH2

Propriedades da mensagem, exceto aquelas contidas no descritor de mensagens (ou extensão) devem ser representadas usando cabeçalhos `MQRFH2`. Isso fornece compatibilidade com a versão anterior para aplicativos que esperam recuperar propriedades, mas não podem ser alterados para usar identificadores de mensagens. Se um `MsgHandle` for fornecido ele será ignorado.

### MQGMO\_PROPERTIES\_COMPATIBILITY

Se a mensagem contiver uma propriedade com um prefixo `"mcd."`, `"jms."`, `"usr."` ou `"mqext."`, todas as propriedades de mensagem serão entregues ao aplicativo em um cabeçalho `MQRFH2`. Caso contrário, todas as propriedades da mensagem, exceto aquelas contidas no descritor de mensagens (ou extensão), serão descartadas e não serão mais acessíveis ao aplicativo.



## Opção Padrão

Se nenhuma das opções descritas for necessária, a seguinte opção poderá ser usada:

### **MQGMO\_NONE**

Use esse valor para indicar que nenhuma outra opção foi especificada. Todas as opções assumem seus valores-padrão. O MQGMO\_NONE auxilia a documentação do programa; não é desejado que essa opção seja usada com nenhuma outra, mas como seu valor é zero, tal uso não pode ser detectado.

O valor inicial do campo Options é MQGMO\_NO\_WAIT mais MQGMO\_PROPERTIES\_AS\_Q\_DEF.

### **WaitInterval (MQLONG) para MQGMO**

Este é o tempo aproximado, expresso em milissegundos, que a chamada MQGET espera que uma mensagem adequada chegue (ou seja, uma mensagem que satisfaça os critérios de seleção especificados no parâmetro **MsgDesc** da chamada MQGET)

**Importante:** Não há espera, ou atraso, se uma mensagem adequada estiver disponível imediatamente.

Consulte o campo *MsgId* descrito em “MQMD - descritor de mensagem” na página 435 para obter mais detalhes) Se nenhuma mensagem adequada tiver chegado após esse tempo decorrido, a chamada será concluída com MQCC\_FAILED e o código de razão MQRC\_NO\_MSG\_AVAILABLE.

No z/OS, o período de tempo que a chamada MQGET realmente espera é afetado pelo carregamento do sistema e considerações de planejamento de trabalho e pode variar entre o valor especificado para *WaitInterval* e aproximadamente 100 milissegundos maiores que *WaitInterval*.

*WaitInterval* é usado junto com a opção MQGMO\_WAIT ou MQGMO\_SET\_SIGNAL. Será ignorado se nenhum deles for especificado. Se um deles for especificado, *WaitInterval* deverá ser maior ou igual a zero ou o valor especial a seguir:

### **MQWI\_UNLIMITED**

Intervalo de espera ilimitado.

O valor inicial desse campo é 0.

### **Signal1 (MQLONG) para MQGMO**

Este é um campo de entrada que é usado apenas em conjunto com a opção MQGMO\_SET\_SIGNAL; ele identifica um sinal que deve ser entregue quando uma mensagem estiver disponível...

**Nota:** O tipo de dados e o uso desse campo são determinados pelo ambiente; por esse motivo, os aplicativos que você deseja portar entre diferentes ambientes não devem usar sinais

- No z/OS, esse campo deve conter o endereço de um Event Control Block (ECB). O BCE deve ser compensado pelo aplicativo antes que a chamada MQGET seja emitida. O armazenamento que contém o BCE não deve ser liberado até que a fila seja fechada. O BCE é postado pelo gerenciador de filas com um dos códigos de conclusão de sinal descritos.. Estes códigos de conclusão são definidos nos bits 2 a 31 do BCE, a área definida na macro IHAECB de mapeamento z/OS como sendo para um código de conclusão do usuário.
- Em todos os outros ambientes, este é um campo reservado; seu valor não é significativo.

Os códigos de conclusão do sinal são:

### **MQEC\_MSG\_CHEGOU**

Uma mensagem adequada chegou na fila.. Esta mensagem não foi reservada ao responsável pela chamada; uma segunda solicitação MQGET deve ser emitida, mas outro aplicativo pode recuperar a mensagem antes da segunda solicitação ser feita.

### **MQEC\_WAIT\_INTERVAL\_EXPIRED**

O *WaitInterval* especificado expirou sem uma mensagem adequada chegando.

### **MQEC\_WAIT\_CANCELADO**

A espera foi cancelada por uma razão indeterminada (como o gerenciador de filas sendo finalizado ou a fila sendo desativada). Emita novamente a solicitação se desejar um diagnóstico adicional.

### **MQEC\_Q\_MGR QUIESCING**

A espera foi cancelada porque o gerenciador de filas entrou no estado de quiesce (MQGMO\_FAIL\_IF QUIESCING foi especificado na chamada MQGET).

### **MQEC\_CONNECTION QUIESCING**

A espera foi cancelada porque a conexão entrou no estado quiesce (MQGMO\_FAIL\_IF QUIESCING foi especificado na chamada MQGET).

O valor inicial desse campo é determinado pelo ambiente:

- No z/OS, o valor inicial é o ponteiro nulo.
- Em todos os outros ambientes, o valor inicial é 0.

### **Signal2 (MQLONG) para MQGMO**

Este é um campo de entrada usado apenas em conjunto com a opção MQGMO\_SET\_SIGNAL. É um campo reservado; seu valor não é significativo.

O valor inicial desse campo é 0.

### **ResolvedQName (MQCHAR48) para MQGMO**

Este é um campo de saída que o gerenciador de fila configura para o nome local da fila da qual a mensagem foi recuperada, conforme definido para o gerenciador de filas locais. Isso é diferente do nome usado para abrir a fila se:

- Uma fila de alias foi aberta (nesse caso, o nome da fila local para a qual o alias resolvido é retornado) ou
- Uma fila modelo foi aberta (nesse caso, o nome da fila local dinâmica é retornado).

O comprimento desse campo é fornecido por MQ\_Q\_NAME\_LENGTH O valor inicial deste campo é a cadeia nula em C e 48 caracteres em branco em outras linguagens de programação.

### **MatchOptions (MQLONG) para MQGMO**

Essas opções permitem que o aplicativo escolha quais campos no parâmetro **MsgDesc** usar para selecionar a mensagem retornada pela chamada MQGET. O aplicativo configura as opções necessárias nesse campo, em seguida, configura os campos correspondentes no parâmetro **MsgDesc** para os valores necessários para esses campos. Apenas as mensagens que possuem esses valores no MQMD para a mensagem são candidatas para recuperação usando esse parâmetro **MsgDesc** na chamada MQGET Os campos para os quais a opção correspondente não é especificada são ignorados ao selecionar a mensagem a ser retornada. Se você especificar nenhum critério de seleção na chamada MQGET (ou seja, qualquer mensagem é aceitável), configure *MatchOptions* para MQMO\_NONE.

- No z/OS, os critérios de seleção que podem ser usados podem ser restritos pelo tipo de índice usado para a fila Consulte o atributo da fila **IndexType** para obter detalhes adicionais

Se você especificar MQGMO\_LOGICAL\_ORDER, apenas determinadas mensagens serão elegíveis para retorno pela próxima chamada MQGET:

- Se não houver nenhum grupo atual ou mensagem lógica, apenas as mensagens que tiverem *MsgSeqNumber* igual a 1 e *Offset* igual a 0 serão elegíveis para retorno. Nessa situação, é possível usar uma ou mais das opções de correspondência a seguir para selecionar qual das mensagens elegíveis é retornado:
  - MQMO\_MATCH\_MSG\_ID
  - MQMO\_MATCH\_CORREL\_ID
  - MQMO\_MATCH\_GROUP\_ID
- Se houver um grupo atual ou mensagem lógica, apenas a próxima mensagem no grupo ou próximo segmento na mensagem lógica será elegível para retorno e isso não poderá ser alterado especificando as opções MQMO\_\*

Em ambos os casos anteriores, é possível especificar opções de correspondência que não se aplicam, mas o valor do campo relevante no parâmetro **MsgDesc** deve corresponder ao valor do

campo correspondente na mensagem a ser retornada; a chamada falha com o código de razão MQRC\_MATCH\_OPTIONS\_ERROR se essa condição não for satisfeita.

*MatchOptions* será ignorado se você especificar MQGMO\_MSG\_UNDER\_CURSOR ou MQGMO\_BROWSE\_MSG\_UNDER\_CURSOR..

Obter mensagens com base na propriedade de mensagem não é feito usando opções de correspondência; para obter mais informações, consulte [“SelectionString \(MQCHARV\) para MQOD” na página 510](#).

É possível especificar uma ou mais das opções de correspondência a seguir:

#### **MQMO\_MATCH\_MSG\_ID**

A mensagem a ser recuperada deve ter um identificador de mensagem que corresponda ao valor do campo *MsgId* no parâmetro **MsgDesc** da chamada MQGET. Essa correspondência é adicional a quaisquer outras correspondências que possam se aplicar (por exemplo, o identificador de correlação).

Se você omitir essa opção, o campo *MsgId* no parâmetro **MsgDesc** será ignorado e qualquer identificador de mensagem será correspondente.

**Nota:** O identificador de mensagem MQMI\_NONE é um valor especial que corresponde a qualquer identificador de mensagem no MQMD da mensagem. Portanto, especificar MQMO\_MATCH\_MSG\_ID com MQMI\_NONE é o mesmo que não especificar MQMO\_MATCH\_MSG\_ID.

#### **MQMO\_MATCH\_CORREL\_ID**

A mensagem a ser recuperada deve ter um identificador de correlações que corresponda ao valor do campo *CorrelId* no parâmetro **MsgDesc** da chamada MQGET. Essa correspondência é adicional a quaisquer outras correspondências que possam se aplicar (por exemplo, o identificador de mensagem).

Se você omitir essa opção, o campo *CorrelId* no parâmetro **MsgDesc** será ignorado e qualquer identificador de correlação será correspondente.

**Nota:** O identificador de correlação MQCI\_NONE é um valor especial que corresponde a *qualquer* identificador de correlação no MQMD para a mensagem. Portanto, especificar MQMO\_MATCH\_CORREL\_ID com MQCI\_NONE é o mesmo que não especificar MQMO\_MATCH\_CORREL\_ID.

#### **MQMO\_MATCH\_GROUP\_ID**

A mensagem a ser recuperada deve ter um identificador de grupo que corresponda ao valor do campo *GroupId* no parâmetro **MsgDesc** da chamada MQGET. Essa correspondência é adicional a quaisquer outras correspondências que possam se aplicar (por exemplo, o identificador de correlação).

Se você omitir essa opção, o campo *GroupId* no parâmetro **MsgDesc** será ignorado e qualquer identificador de grupo será correspondente.

**Nota:** O identificador de grupo MQGI\_NONE é um valor especial que corresponde a *qualquer* identificador de grupo no MQMD da mensagem. Portanto, especificar MQMO\_MATCH\_GROUP\_ID com MQGI\_NONE é o mesmo que não especificar MQMO\_MATCH\_GROUP\_ID.

#### **MQMO\_MATCH\_MSG\_SEQ\_NUMBER**

A mensagem a ser recuperada deve ter um número de sequência de mensagem que corresponda ao valor do campo *MsgSeqNumber* no parâmetro **MsgDesc** da chamada MQGET.. Essa correspondência é além de quaisquer outras correspondências que possam se aplicar (por exemplo, o identificador de grupo).

Se você omitir essa opção, o campo *MsgSeqNumber* no parâmetro **MsgDesc** será ignorado, e qualquer número de sequência de mensagem será correspondente

#### **MQMO\_MATCH\_OFFSET**

A mensagem a ser recuperada deve ter um deslocamento que corresponda ao valor do campo *Offset* no parâmetro **MsgDesc** da chamada MQGET. Essa correspondência é além de quaisquer outras correspondências que possam se aplicar (por exemplo, o número de sequência da mensagem).

Se você omitir essa opção, o campo *Offset* no parâmetro **MsgDesc** será ignorado e qualquer deslocamento será correspondente.

- Essa opção não é suportada no z/OS

### **MQMO\_MATCH\_MSG\_TOKEN**

A mensagem a ser recuperada deve ter um token de mensagem que corresponda ao valor do campo *MsgToken* na estrutura MQGMO especificada na chamada MQGET..

É possível especificar essa opção para todas as filas locais. Se você especificá-lo para uma fila que tenha um *IndexType* de MQIT\_MSG\_TOKEN (uma fila gerenciada pelo WLM), não será possível especificar nenhuma outra opção de correspondência com MQMO\_MATCH\_MSG\_TOKEN...

Não é possível especificar MQMO\_MATCH\_MSG\_TOKEN com MQGMO\_WAIT ou MQGMO\_SET\_SIGNAL. Se o aplicativo desejar aguardar uma mensagem chegar em uma fila que tenha um *IndexType* de MQIT\_MSG\_TOKEN, especifique MQMO\_NONE.

Se você omitir essa opção, o campo *MsgToken* em MQGMO será ignorado, e qualquer token de mensagem será correspondente.

Se você não especificar nenhuma das opções descritas, poderá usar a opção a seguir:

### **MQMO\_NONE**

Não use nenhuma correspondência ao selecionar a mensagem a ser retornada; todas as mensagens na fila são elegíveis para recuperação (mas sujeitas ao controle pelas opções MQGMO\_ALL\_MSGS\_AVAILABLE, MQGMO\_ALL\_SEGMENTS\_AVAILABLE, e MQGMO\_COMPLETE\_MSG).

MQMO\_NONE ajuda a documentação do programa. Não se pretende que essa opção seja usada com qualquer outra opção MQMO\_\*, mas como seu valor é zero, tal uso não pode ser detectado.

Este é um campo de entrada. O valor inicial desse campo é MQMO\_MATCH\_MSG\_ID com MQMO\_MATCH\_CORREL\_ID. Este campo será ignorado se *Version* for menor que MQGMO\_VERSION\_2.

**Nota:** O valor inicial do campo *MatchOptions* é definido para compatibilidade com gerenciadores de filas anteriores do MQSeries. No entanto, ao ler uma série de mensagens de uma fila sem usar critérios de seleção, esse valor inicial requer que o aplicativo reconfigure os campos *MsgId* e *CorrelId* para MQMI\_NONE e MQCI\_NONE antes de cada chamada MQGET. Evite a necessidade de reconfigurar *MsgId* e *CorrelId* configurando *Version* para MQGMO\_VERSION\_2 e *MatchOptions* para MQMO\_NONE.

### **Conceitos relacionados**

[Seletores de mensagens no JMS..](#)

### **GroupStatus (MQCHAR) para MQGMO.**

Esse sinalizador indica se a mensagem recuperada está em um grupo.

Ele possui um dos seguintes valores:

#### **MQGS\_NOT\_IN\_GROUP**

A mensagem não está em um grupo.

#### **MQGS\_MSG\_IN\_GROUP**

A mensagem está em um grupo, mas não é a última no grupo.

#### **MQGS\_LAST\_MSG\_IN\_GROUP**

A mensagem é a última no grupo.

Este também é o valor retornado se o grupo consistir em apenas uma mensagem.

Esse é um campo de saída. O valor inicial desse campo é MQGS\_NOT\_IN\_GROUP. Este campo será ignorado se *Version* for menor que MQGMO\_VERSION\_2.

### **SegmentStatus (MQCHAR) para MQGMO.**

Este é um sinalizador que indica se a mensagem recuperada é um segmento de uma mensagem lógica. Ele possui um dos seguintes valores:

### **MQSS\_NOT\_A\_SEGMENT**

A mensagem não é um segmento.

### **MQSS\_SEGMENT**

A mensagem é um segmento, mas não é o último segmento da mensagem lógica.

### **MQSS\_LAST\_SEGMENT**

A mensagem é o último segmento da mensagem lógica.

Este também é o valor retornado se a mensagem lógica consistir em apenas um segmento.

No z/OS, o gerenciador de filas sempre configura esse campo para MQSS\_NOT\_A\_SEGMENT..

Esse é um campo de saída. O valor inicial desse campo é MQSS\_NOT\_A\_SEGMENT. Este campo será ignorado se *Version* for menor que MQGMO\_VERSION\_2.

### **Segmentação (MQCHAR) para MQGMO**

Este é um sinalizador que indica se a segmentação adicional é permitida para a mensagem recuperada. Ele possui um dos seguintes valores:

#### **MQSEG\_INIBIDA**

Segmentação não permitida.

#### **MQSEG\_ALLOWED**

Segmentação permitida.

No z/OS, o gerenciador de filas sempre configura esse campo para MQSEG\_INHIBITED

Esse é um campo de saída. O valor inicial desse campo é MQSEG\_IBITED. Este campo será ignorado se *Version* for menor que MQGMO\_VERSION\_2.

### **Reserved1 (MQCHAR) para MQGMO**

Este é um campo reservado. O valor inicial desse campo é um caractere em branco. Este campo será ignorado se *Version* for menor que MQGMO\_VERSION\_2.

### **MsgToken (MQBYTE16) para MQGMO**

Campo MsgToken -Estrutura de MQGMO. Esse campo é usado pelo gerenciador de filas para identificar exclusivamente uma mensagem..

Essa é uma sequência de bytes gerada pelo gerenciador de filas para identificar uma mensagem exclusivamente em uma fila. O token de mensagem é gerado quando a mensagem é colocada pela primeira vez no gerenciador de filas e permanece com a mensagem até que a mensagem seja removida permanentemente do gerenciador de filas, a menos que o gerenciador de filas seja reiniciado.

Quando a mensagem é removida da fila, o *MsgToken* que identificou essa instância da mensagem não é mais válido, e nunca é reutilizado. Se o gerenciador de filas for reiniciado, o *MsgToken* que identificou uma mensagem na fila antes da reinicialização poderá não ser válido após a reinicialização. No entanto, o *MsgToken* nunca é reutilizado para identificar uma instância de mensagem diferente. O *MsgToken* é gerado pelo gerenciador de filas e não é visível para nenhum aplicativo externo.

Quando uma mensagem é retornada por uma chamada para MQGET em que um MQGMO Versão 3 ou superior é fornecido, o *MsgToken* que identifica a mensagem na fila é retornado no MQGMO pelo gerenciador de filas. Há uma exceção para isso: quando a mensagem está sendo removida da fila fora do ponto de sincronização, o gerenciador de fila pode não retornar um *MsgToken* porque não é útil identificar a mensagem retornada em uma chamada MQGET subsequente. Os aplicativos devem usar apenas *MsgToken* para referir-se à mensagem em chamadas MQGET subsequentes.

Se um *MsgToken* for fornecido e o *MatchOption* MQMO\_MATCH\_MSG\_TOKEN for especificado e nem MQGMO\_MSG\_UNDER\_CURSOR nem MQGMO\_BROWSE\_MSG\_UNDER\_CURSOR for especificado, somente a mensagem identificada por esse *MsgToken* poderá ser retornada. A opção é válida em todas as filas locais, independentemente de INDXTYPE, e em z/OS deve-se usar INDXTYPE (MSGTOKEN) somente em filas do Workload Manager (WLM).

Qualquer outro *MatchOptions* especificado é verificado e, se não corresponder, MQRC\_NO\_MSG\_AVAILABLE será retornado. Se MQGMO\_BROWSE\_NEXT for codificado com MQMO\_MATCH\_MSG\_TOKEN, a mensagem identificada pelo *MsgToken* será retornada somente se ela estiver além do cursor de navegação para a manipulação de chamadas

MQGMO\_MSG\_UNDER\_CURSOR ou MQGMO\_BROWSE\_MSG\_UNDER\_CURSOR for especificado, MQMO\_MATCH\_MSG\_TOKEN será ignorado.

MQMO\_MATCH\_MSG\_TOKEN não é válido com as seguintes opções de mensagem get:

- MQGMO\_WAIT
- MQGMO\_SET\_SIGNAL

Para uma chamada MQGET especificando MQMO\_MATCH\_MSG\_TOKEN, um MQGMO da versão 3 ou posterior deve ser fornecido para a chamada, caso contrário, MQRC\_ERR\_GMO\_VERSION será retornado.

Se *MsgToken* não for válido neste momento, MQCC\_FAILED com MQRC\_NO\_MSG\_AVAILABLE será retornado, a menos que haja outro erro.

### ***ReturnedLength (MQLONG) para MQGMO***

Esse é um campo de saída que o gerenciador de fila configura para o comprimento em bytes dos dados da mensagem retornados pela chamada MQGET no parâmetro **Buffer**. Se o gerenciador de filas não suportar esse recurso, *ReturnedLength* será configurado para o valor MQRL\_UNDEFINED.

Quando as mensagens são convertidas entre codificações ou conjuntos de caracteres, os dados da mensagem às vezes podem mudar de tamanho. No retorno da chamada MQGET:

- Se o *ReturnedLength* não for MQRL\_UNDEFINED, o número de bytes de dados da mensagem retornados será fornecido pelo *ReturnedLength*
- Se *ReturnedLength* tiver o valor MQRL\_UNDEFINED, o número de bytes de dados da mensagem retornados geralmente será fornecido pelo menor de *BufferLength* e *DataLength*, mas poderá ser *menor que* se a chamada MQGET for concluída com o código de razão MQRC\_TRUNCATED\_MSG\_ACCEPTED. Se isso acontecer, os bytes insignificantes no parâmetro **Buffer** serão configurados como nulos

O seguinte valor especial é definido:

#### **MQRL\_UNDEFINED**

Comprimento de dados retornados não definido.

No z/OS, o valor retornado para o campo *ReturnedLength* é sempre MQRL\_UNDEFINED.

O valor inicial desse campo é MQRL\_UNDEFINED. Este campo será ignorado se *Version* for menor que MQGMO\_VERSION\_3.

### ***Reserved2 (MQLONG) para MQGMO***

Este é um campo reservado. O valor inicial desse campo é um caractere em branco. Esse campo será ignorado, se *Version* for menor que MQGMO\_VERSION\_4

### ***MsgHandle (MQHMSG) para MQGMO***

Se a opção MQGMO\_PROPERTIES\_AS\_Q\_DEF for especificada e o atributo da fila **PropertyControl** não for configurado como MQPROP\_FORCE\_MQRFH2, esta será a manipulação para uma mensagem que será preenchida com as propriedades da mensagem que está sendo recuperada da fila... O identificador é criado por uma chamada MQCRTM. Quaisquer propriedades já associadas ao identificador serão limpas antes de recuperar uma mensagem.

O seguinte valor também pode ser especificado:

MQHM\_NONE

Nenhuma manipulação de mensagem fornecida.

Nenhum descritor de mensagens é necessário na chamada MQGET se um identificador de mensagens válido for fornecido e usado na saída para conter as propriedades de mensagem, o descritor de mensagens associado ao identificador de mensagens é usado para campos de entrada

Se um descritor de mensagens for especificado na chamadas MQGET, ele sempre terá precedência sobre o descritor de mensagem associado a um identificador de mensagens

Se MQGMO\_PROPERTIES\_FORCE\_MQRFH2 for especificado ou MQGMO\_PROPERTIES\_AS\_Q\_DEF for especificado e o atributo da fila **PropertyControl** for MQPROP\_FORCE\_MQRFH2 , a chamada falhará com o código de razão MQRC\_MD\_ERROR quando nenhum parâmetro do descritor de mensagens for especificado.

No retorno da chamada MQGET, as propriedades e o descritor de mensagens associados a essa manipulação de mensagens são atualizados para refletir o estado da mensagem recuperada (bem como o descritor de mensagens se um foi fornecido na chamada MQGET). As propriedades da mensagem podem então ser consultadas usando a chamada MQINQMP.

Exceto para as extensões do descritor de mensagens, quando presentes, uma propriedade que pode ser consultada com a chamada MQINQMP não está contida nos dados da mensagem; se a mensagem na fila continha propriedades nos dados da mensagem, eles serão removidos dos dados da mensagem antes que os dados sejam retornados ao aplicativo.

Se nenhuma manipulação de mensagem for fornecida, ou a Versão for menor que MQGMO\_VERSION\_4 , você deverá fornecer um descritor de mensagens válido na chamada MQGET Quaisquer propriedades de mensagens (exceto aquelas contidas no descritor de mensagens) são retornadas nos dados da mensagem sujeitos ao valor das opções de propriedades na estrutura MQGMO e no atributo da fila **PropertyControl** .

Este é um campo de entrada. O valor inicial desse campo é MQHM\_NONE. Esse campo será ignorado se **Version** for menor que MQGMO\_VERSION\_4.

## **MQIIH - Informações de cabeçalho do IMS**

A estrutura MQIIH descreve as informações do cabeçalho para uma mensagem enviada para o IMS através da ponte IMSPara qualquer plataforma suportada pelo IBM MQ , é possível criar e transmitir uma mensagem que inclui a estrutura MQIIH, mas apenas um gerenciador de filas do IBM MQ for z/OS pode usar a ponte IMS Portanto, para que a mensagem chegue ao IMS a partir de um gerenciador de filas nãoz/OS , sua rede do gerenciador de filas deve incluir pelo menos um gerenciador de filas do z/OS por meio do qual a mensagem pode ser roteada

### **Disponibilidade**

All IBM MQ systems and IBM MQ clients.

### **Nome do Formato**

MQFMT\_IMS

### **Conjunto de caracteres e codificação**

Condições especiais se aplicam ao conjunto de caracteres e à codificação usados para a estrutura MQIIH e dados da mensagem do aplicativo:

- Aplicativos que se conectam ao gerenciador de filas que possui a fila de ponte IMS devem fornecer uma estrutura MQIIH que esteja no conjunto de caracteres e na codificação do gerenciador de filas. Isso ocorre porque a conversão de dados da estrutura MQIIH não é executada neste caso

- Os aplicativos que se conectam a outros gerenciadores de filas podem fornecer uma estrutura MQIIH que esteja em qualquer um dos conjuntos de caracteres e codificações suportados; o agente do canal de mensagem de recebimento conectado ao gerenciador de filas que possui a fila de ponte IMS converte o MQIIH.
- Os dados da mensagem do aplicativo após a estrutura MQIIH devem estar no mesmo conjunto de caracteres e codificação que a estrutura MQIIH. Não use os campos *CodedCharSetId* e *Encoding* na estrutura MQIIH para especificar o conjunto de caracteres e a codificação dos dados da mensagem do aplicativo

Deve-se fornecer uma saída de conversão de dados para converter os dados da mensagem do aplicativo se os dados não forem um dos formatos integrados suportados pelo gerenciador de filas

## Campos

**Nota:** Na tabela a seguir, os campos são agrupados por uso e não alfabeticamente. Os tópicos filhos seguem a mesma sequência.

*Tabela 497. Campos em MQIIH para MQIIH*

Nome e descrição do campo	Nome da constante	Valor inicial (se houver) da constante
<u>StrucId</u> (identificador de estrutura)	MQIIH_STRUC_ID	' I IH↵ '
<u>Versão</u> (número da versão da estrutura).	MQIIH_VERSION_1	1
<u>StrucLength</u> (comprimento da estrutura MQIIH)	MQIIH_LENGTH_1	84
<u>Codificação</u> (reservado-consulte “Conjunto de caracteres e codificação” na página 419)	Nenhum	0
<u>CodedCharSetId</u> (reservado-consulte “Conjunto de caracteres e codificação” na página 419)	Nenhum	0
<u>Formato</u> (nome do formatoMQ de dados que seguem MQIIH)	MQFMT_NONE	Espaços em branco
<u>Sinalizadores</u> (sinalizadores)	MQIIH_NONE	0
<u>LTermOverride</u> (substituição de terminal lógico)	Nenhum	Espaços em branco
<u>MFSMapName</u> (nome do mapeamento de serviços de formato de mensagem).	Nenhum	Espaços em branco
<u>ReplyToFormat</u> (nome do formato da mensagem de resposta doMQ )	MQFMT_NONE	Espaços em branco
<u>Autenticador</u> (RACF senha ou passticket)	MQIAUT_NONE	Espaços em branco
<u>TranInstanceId</u> (identificador da instância de transação)	MQITII_NONE	Nulos
<u>TranState</u> (estado da transação)	MQITS_NOT_IN_CONVE RSATION	' ↵ '
<u>CommitMode</u> (modo de confirmação)	MQICM_COMMIT_THEN _SEND	' 0 '
<u>SecurityScope</u> (escopo de segurança)	MQISS_CHECK	' C '
<u>Reservado</u> (reservado).	Nenhum	' ↵ '



Tabela 497. Campos em MQIIH para MQIIH (continuação)

Nome e descrição do campo	Nome da constante	Valor inicial (se houver) da constante
<b>Notas:</b>		
<p>1. O símbolo ~ representa um único caractere em branco.</p> <p>2. Na linguagem de programação C, a variável macroMQIIH_DEFAULT contém os valores listados na tabela. Ele pode ser usado da seguinte maneira para fornecer valores iniciais para os campos na estrutura:</p>		
<pre>MQIIH MyIIH = {MQIIH_DEFAULT};</pre>		

## Declarações de idiomas

### Declaração C para MQIIH

```
typedef struct tagMQIIH MQIIH;
struct tagMQIIH {
    MQCHAR4   StrucId;           /* Structure identifier */
    MQLONG    Version;          /* Structure version number */
    MQLONG    StrucLength;      /* Length of MQIIH structure */
    MQLONG    Encoding;        /* Reserved */
    MQLONG    CodedCharSetId;   /* Reserved */
    MQCHAR8   Format;           /* MQ format name of data that follows
                                MQIIH */
    MQLONG    Flags;            /* Flags */
    MQCHAR8   LTermOverride;    /* Logical terminal override */
    MQCHAR8   MFSMapName;      /* Message format services map name */
    MQCHAR8   ReplyToFormat;    /* MQ format name of reply message */
    MQCHAR8   Authenticator;    /* RACF password or passticket */
    MQBYTE16  TranInstanceId;   /* Transaction instance identifier */
    MQCHAR    TranState;        /* Transaction state */
    MQCHAR    CommitMode;       /* Commit mode */
    MQCHAR    SecurityScope;    /* Security scope */
    MQCHAR    Reserved;        /* Reserved */
};
```

### Declaração COBOL para MQIIH

```
** MQIIH structure
10 MQIIH.
** Structure identifier
15 MQIIH-STRUCID PIC X(4).
** Structure version number
15 MQIIH-VERSION PIC S9(9) BINARY.
** Length of MQIIH structure
15 MQIIH-STRUCLength PIC S9(9) BINARY.
** Reserved
15 MQIIH-ENCODING PIC S9(9) BINARY.
** Reserved
15 MQIIH-CODEDCHARSETID PIC S9(9) BINARY.
** MQ format name of data that follows MQIIH
15 MQIIH-FORMAT PIC X(8).
** Flags
15 MQIIH-FLAGS PIC S9(9) BINARY.
** Logical terminal override
15 MQIIH-LTERMOverride PIC X(8).
** Message format services map name
15 MQIIH-MFSMAPNAME PIC X(8).
** MQ format name of reply message
15 MQIIH-REPLYTOFORMAT PIC X(8).
** RACF password or passticket
15 MQIIH-AUTHENTICATOR PIC X(8).
** Transaction instance identifier
15 MQIIH-TRANINSTANCEID PIC X(16).
** Transaction state
15 MQIIH-TRANSTATE PIC X.
** Commit mode
```

```

15 MQIIH-COMMITMODE      PIC X.
** Security scope
15 MQIIH-SECURITYSCOPE  PIC X.
** Reserved
15 MQIIH-RESERVED       PIC X.

```

### Declaração PL/I para MQIIH

```

dcl
1 MQIIH based,
3 StrucId      char(4),          /* Structure identifier */
3 Version      fixed bin(31),   /* Structure version number */
3 StrucLength  fixed bin(31),   /* Length of MQIIH structure */
3 Encoding     fixed bin(31),   /* Reserved */
3 CodedCharSetId fixed bin(31), /* Reserved */
3 Format        char(8),         /* MQ format name of data that follows
                               MQIIH */
3 Flags        fixed bin(31),   /* Flags */
3 LTermOverride char(8),       /* Logical terminal override */
3 MFSSMapName  char(8),        /* Message format services map name */
3 ReplyToFormat char(8),       /* MQ format name of reply message */
3 Authenticator char(8),       /* RACF password or passticket */
3 TranInstanceId char(16),     /* Transaction instance identifier */
3 TranState    char(1),        /* Transaction state */
3 CommitMode   char(1),        /* Commit mode */
3 SecurityScope char(1),       /* Security scope */
3 Reserved     char(1);        /* Reserved */

```

### Declaração High Level Assembler para MQIIH

```

MQIIH          DSECT
MQIIH_STRUCID  DS   CL4   Structure identifier
MQIIH_VERSION  DS   F     Structure version number
MQIIH_STRUCLNGTH DS  F     Length of MQIIH structure
MQIIH_ENCODING DS   F     Reserved
MQIIH_CODEDCHARSETID DS  F     Reserved
MQIIH_FORMAT   DS   CL8   MQ format name of data that follows
*
MQIIH_FLAGS    DS   F     Flags
MQIIH_LTERM_OVERRIDE DS  CL8 Logical terminal override
MQIIH_MFSMAPNAME DS  CL8  Message format services map name
MQIIH_REPLYTOFORMAT DS  CL8 MQ format name of reply message
MQIIH_AUTHENTICATOR DS  CL8 RACF password or passticket
MQIIH_TRANINSTANCEID DS  XL16 Transaction instance identifier
MQIIH_TRANSTATE DS   CL1  Transaction state
MQIIH_COMMITMODE DS   CL1  Commit mode
MQIIH_SECURITYSCOPE DS  CL1 Security scope
MQIIH_RESERVED DS   CL1  Reserved
*
MQIIH_LENGTH   EQU  *-MQIIH
ORG  MQIIH
MQIIH_AREA     DS   CL(MQIIH_LENGTH)

```

### Declaração Visual Basic para MQIIH

```

Type MQIIH
StrucId      As String*4 'Structure identifier'
Version      As Long     'Structure version number'
StrucLength  As Long     'Length of MQIIH structure'
Encoding     As Long     'Reserved'
CodedCharSetId As Long   'Reserved'
Format       As String*8 'MQ format name of data that follows MQIIH'
Flags        As Long     'Flags'
LTermOverride As String*8 'Logical terminal override'
MFSSMapName  As String*8 'Message format services map name'
ReplyToFormat As String*8 'MQ format name of reply message'
Authenticator As String*8 'RACF password or passticket'
TranInstanceId As MQBYTE16 'Transaction instance identifier'
TranState    As String*1 'Transaction state'
CommitMode   As String*1 'Commit mode'
SecurityScope As String*1 'Security scope'
Reserved     As String*1 'Reserved'
End Type

```

### ***StrucId (MQCHAR4) para MQIIH***

Este é o identificador de estrutura da estrutura do cabeçalho de informações IMS . Ele é sempre um campo de entrada Seu valor é MQIIH\_STRUC\_ID.

O valor deve ser:.

#### **MQIIH\_STRUC\_ID**

Identificador para a estrutura do cabeçalho de informações IMS .

Para a linguagem de programação C, a constante MQIIH\_STRUC\_ID\_ARRAY também é definida. Ele tem o mesmo valor que MQIIH\_STRUC\_ID, mas é uma matriz de caracteres em vez de uma cadeia.

### ***Versão (MQLONG) para MQIIH***

Este é o número da versão da estrutura O valor deve ser:.

#### **MQIIH\_VERSION\_1**

Número da versão para a estrutura do cabeçalho de informações do IMS

A constante a seguir especifica o número da versão atual:

#### **MQIIH\_CURRENT\_VERSION**

Versão atual da estrutura do cabeçalho de informações IMS .

O valor inicial desse campo é MQIIH\_VERSION\_1.

### ***StrucLength (MQLONG) para MQIIH***

É o comprimento da estrutura MQIIH. O valor deve ser:.

#### **MQIIH\_LENGTH\_1**

Comprimento da estrutura do cabeçalho de informações do IMS

O valor inicial desse campo é MQIIH\_LENGTH\_1.

### ***Codificação (MQLONG) para MQIIH***

Este é um campo reservado; seu valor não é significativo. O valor inicial desse campo é 0.

A Codificação para estruturas suportadas que seguem uma estrutura MQIIH é a mesma da própria estrutura MQIIH e obtida de qualquer cabeçalho anterior do MQ .

### ***CodedCharSetId (MQLONG) para MQIIH***

Este é um campo reservado; seu valor não é significativo. O valor inicial desse campo é 0.

O ID do conjunto de caracteres para estruturas suportadas que seguem uma estrutura MQIIH é o mesmo que aquele da própria estrutura MQIIH e obtido de qualquer cabeçalho MQ anterior.

### ***Formato (MQCHAR8) para MQIIH***

Especifica o nome do formato MQ dos dados que seguem a estrutura MQIIH.

Na chamada MQPUT ou MQPUT1 , o aplicativo deve configurar esse campo para o valor apropriado aos dados.

O comprimento desse campo é fornecido por MQ\_FORMAT\_LENGTH O valor inicial desse campo é MQFMT\_NONE.

### ***Sinalizadores (MQLONG) para MQIIH***

O valor dos sinalizadores deve ser:

#### **MQIIH\_NONE**

Sem bandeiras.

#### **MQIIH\_PASS\_EXPIRATION**

A mensagem de resposta contém:

- As mesmas opções de relatório de expiração que a mensagem de solicitação
- O tempo de expiração restante da mensagem de pedido sem ajuste feito para o tempo de processamento da ponte

Se esse valor não for configurado, o tempo de expiração será configurado como *unlimited*.

#### **MQIIH\_REPLY\_FORMAT\_NONE**

Configura o MQIIH do MQIIH.Format da resposta para MQFMT\_NONE.

#### **MQIIH\_IGNORE\_PURG**

Configura o indicador TMAMIPRG no prefixo OTMA, que solicita que o OTMA ignore chamadas PURG no PCB TP para transações CMO .

#### **MQIIH\_CMO\_REQUEST\_RESPONSE**

Para transações do Modo de Confirmação 0 (CMO), esse sinalizador configura o indicador TMAMHRSP no prefixo OTMA. A configuração desse indicador solicita que OTMA/IMS gere uma mensagem DFS2082 RESPONSE MODE TRANSACTION FINALIZADO SEM RESPOSTA quando o programa de aplicativo IMS original não responde ao IOPCB nem troca de mensagem para outra transação.

O valor inicial desse campo é MQIIH\_NONE.

#### ***LTermOverride (MQCHAR8) para MQIIH***

A substituição do terminal lógico, colocada no campo PCB de E/S. É opcional; se não for especificado, o nome do TPIPE será usado. Ele será ignorado se o primeiro byte estiver em branco ou nulo.

O comprimento desse campo é fornecido por MQ\_LTERM\_OVERRIDE\_LENGTH O valor inicial deste campo é 8 caracteres em branco.

#### ***MFSMapName (MQCHAR8) para MQIIH***

O nome do mapa de serviços de formato de mensagem, colocado no campo PCB de E/S. Ele é opcional. Na entrada ele representa o MID, na saída ele representa o MOD. Ele será ignorado se o primeiro byte estiver em branco ou nulo.

O comprimento desse campo é fornecido por MQ\_MFS\_MAP\_NAME\_LENGTH O valor inicial deste campo é 8 caracteres em branco.

#### ***ReplyToFormat (MQCHAR8) para MQIIH***

Esse é o nome do formato do MQ da mensagem de resposta enviada em resposta à mensagem atual. O comprimento desse campo é fornecido por MQ\_FORMAT\_LENGTH O valor inicial desse campo é MQFMT\_NONE.

Para converter os dados na mensagem de resposta usando MQGMO\_CONVERT, especifique MQIIH.replyToFormat= MQFMT\_STRING ou MQIIH.replyToFormat= MQFMT\_IMS\_VAR\_STRING. Para obter uma explicação do uso desses campos, consulte [“Formato \(MQCHAR8\) para MQMD” na página 462](#).

Se o valor padrão (MQIIH.replyToFormat= MQFMT\_NONE) for usado na mensagem de solicitação e a mensagem de resposta for recuperada usando MQGMO\_CONVERT, nenhuma conversão de dados será executada.

#### ***Autenticador (MQCHAR8) para MQIIH***

Esta é a RACF senha ou PassTicket Ele é opcional; se especificado, será usado com o ID do usuário no contexto de segurança do MQMD para construir um UTOKEN que é enviado para o IMS para fornecer um contexto de segurança. Se ele não for especificado, o ID do usuário será usado sem verificação. Isso depende da configuração dos comutadores RACF , que podem requerer que um autenticador esteja presente..

Isso será ignorado se o primeiro byte estiver em branco ou nulo. O seguinte valor especial pode ser usado:

## **MQIAUT\_NONE**

Nenhuma autenticação.

Para a linguagem de programação C, a constante MQIAUT\_NONE\_ARRAY também é definida; ela tem o mesmo valor que MQIAUT\_NONE, mas é uma matriz de caracteres em vez de uma sequência.

O comprimento desse campo é fornecido por MQ\_AUTHENTICATOR\_LENGTH O valor inicial desse campo é MQIAUT\_NONE.

## **ID TranInstance(MQBYTE16) para MQIIH**

Esse é o identificador da instância de transação Esse campo é usado pelas mensagens de saída de IMS, portanto, é ignorado na primeira entrada Se você configurar *TranState* como MQITS\_IN\_CONVERSATION, isso deverá ser fornecido na próxima entrada e em todas as entradas subsequentes, para permitir que o IMS correlacione as mensagens com a conversa correta É possível usar o seguinte valor especial:

### **MQITII\_NONE**

Nenhum identificador de instância de transação.

Para a linguagem de programação C, a constante MQITII\_NONE\_ARRAY também é definida; isso tem o mesmo valor que MQITII\_NONE, mas é uma matriz de caracteres em vez de uma cadeia.

O comprimento desse campo é fornecido por MQ\_TRAN\_INSTANCE\_ID\_LENGTH O valor inicial desse campo é MQITII\_NONE.

## **TranState (MQCHAR) para MQIIH**

Indica o estado da conversa IMS . Isso é ignorado na primeira entrada porque não existe nenhuma conversa. Em entradas subsequentes, isso indica se uma conversa está ativa ou não; Na saída, ele é configurado por IMS O valor deve ser um dos seguintes:

### **MQITS\_IN\_CONVERSATION**


Em conversa.

### **MQITS\_NOT\_IN\_CONVERSATION**

Não na conversa.

### **MQITS\_ARCHITECTED**

Retornar dados de estado da transação no formato arquitetado.

Esse valor é usado somente com o comando IMS /DISPLAY TRAN .. Ele retorna os dados de estado da transação no formato arquitetado IMS em vez de no formato de caractere  Para obter mais informações, consulte [Gravando IMS programas de transação por IBM MQ](#).

O valor inicial desse campo é MQITS\_NOT\_IN\_CONVERSATION.

## **CommitMode (MQCHAR) para MQIIH**

Este é o modo de confirmação do IMS Consulte o *OTMA Reference* para obter mais informações sobre os modos de confirmação IMS . O valor deve ser um dos seguintes:

### **MQICM\_COMMIT\_THEN\_SEND**

Confirmar, em seguida, enviar

Esse modo implica em enfileiramento duplo de tempos de ocupação de região de saída, mas mais curtos. Transações de atalho e conversa não podem ser executadas com esse modo.

### **MQICM\_SEND\_THEN\_COMMIT**

Enviar e, em seguida, confirmar

Qualquer transação IMS iniciada como resultado de um modo de confirmação de MQICM\_SEND\_THEN\_COMMIT é executada no modo RESPONSE independentemente de como a transação é definida na definição do sistema IMS (parâmetro MSGTYPE na macro TRANSACT). Isso também se aplica a transações iniciadas por meio de uma comutação de transação

O valor inicial deste campo é MQICM\_COMMIT\_THEN\_SEND.

## SecurityScope (MQCHAR) para MQIIH

Isso indica o processamento de segurança IMS necessário. Os seguintes valores são definidos:

### MQISS\_CHECK

Verificar escopo de segurança: um ACEE é construído na região de controle, mas não na região dependente.

### MQISS\_FULL

Escopo de segurança total: um ACEE em cache é construído na região de controle e um ACEE não em cache é construído na região dependente. Se você usar MQISS\_FULL, assegure que o ID do usuário para o qual o ACEE é construído tenha acesso aos recursos usados na região dependente.

Se nem MQISS\_CHECK nem MQISS\_FULL for especificado para esse campo, MQISS\_CHECK será assumido

O valor inicial desse campo é MQISS\_CHECK.

## Reservado (MQCHAR) para MQIIH

Este é um campo reservado; ele deve estar em branco

## MQIMPO-Consultar opções de propriedade de mensagem

A estrutura MQIMPO permite que aplicativos especifiquem opções que controlam como as propriedades de mensagens são consultadas. A estrutura é um parâmetro de entrada na chamada MQINQMP..

### Disponibilidade

All IBM MQ systems and IBM MQ clients.

### Conjunto de caracteres e codificação

Os dados em MQIMPO devem estar no conjunto de caracteres do aplicativo e na codificação do aplicativo (MQENC\_NATIVE)..

### Campos

**Nota:** Na tabela a seguir, os campos são agrupados por uso e não alfabeticamente. Os tópicos filhos seguem a mesma sequência.

Nome e descrição do campo	Nome da constante	Valor inicial (se houver) da constante
StrucId (identificador de estrutura)	MQIMPO_STRUC_ID	'IMPO'
Versão (número da versão da estrutura).	MQIMPO_VERSION_1	1
Opções (opções controlando a ação MQINQMP)	MQIMPO_INQ_FIRST	
RequestedEncoding (codificação na qual a propriedade consultada deve ser convertida)	MQENC_NATIVE	
RequestedCCSID (conjunto de caracteres da propriedade solicitada)	MQCCSI_APPL	
ReturnedEncoding (codificação do valor retornado)	MQENC_NATIVE	
ReturnedCCSID	0	
Reserved1 (campo reservado)	caractere em branco (campo de 4 bytes)	
ReturnedName (nome da propriedade consultada)	MQCHARV_DEFAULT	

Tabela 498. Campos em MQIPMO (continuação)

Nome e descrição do campo	Nome da constante	Valor inicial (se houver) da constante
TypeString (representação em sequência do tipo de dados da propriedade)	Cadeia nula ou espaços em branco	

**Notas:**

1. O valor Sequência nula ou espaços em branco indica a sequência nula em C e caracteres em branco em outras linguagens de programação.
2. Na linguagem de programação C, a variável macroMQIMPO\_DEFAULT contém os valores listados na tabela. Use-o da seguinte maneira para fornecer valores iniciais para os campos na estrutura:

```
MQIMPO MyIMPO = {MQIMPO_DEFAULT};
```

## Declarações de idiomas

### Declaração C para MQIMPO

```
typedef struct tagMQIMPO MQIMPO;
struct tagMQIMPO {
    MQCHAR4  StrucId;           /* Structure identifier */
    MQLONG   Version;          /* Structure version number */
    MQLONG   Options;          /* Options that control the action of
                               MQINQMP */
    MQLONG   RequestedEncoding; /* Requested encoding of Value */
    MQLONG   RequestedCCSID;   /* Requested character set identifier
                               of Value */
    MQLONG   ReturnedEncoding; /* Returned encoding of Value */
    MQLONG   ReturnedCCSID;    /* Returned character set identifier
                               of Value */
    MQCHAR   Reserved1;        /* Reserved field */
    MQCHARV  ReturnedName;     /* Returned property name */
    MQCHAR8  TypeString;       /* Property data type as a string */
};
```

### Declaração COBOL para MQIMPO

```
** MQIMPO structure
10 MQIMPO.
** Structure identifier
15 MQIMPO-STRUCID          PIC X(4).
** Structure version number
15 MQIMPO-VERSION        PIC S9(9) BINARY.
** Options that control the action of MQINQMP
15 MQIMPO-OPTIONS        PIC S9(9) BINARY.
** Requested encoding of VALUE
15 MQIMPO-REQUESTEDENCODING PIC S9(9) BINARY.
** Requested character set identifier of VALUE
15 MQIMPO-REQUESTEDCCSID  PIC S9(9) BINARY.
** Returned encoding of VALUE
15 MQIMPO-RETURNEDENCODING PIC S9(9) BINARY.
** Returned character set identifier of VALUE
15 MQIMPO-RETURNEDCCSID  PIC S9(9) BINARY.
** Reserved field
15 MQIMPO-RESERVED1
** Returned property name
15 MQIMPO-RETURNEDNAME.
** Address of variable length string
20 MQIMPO-RETURNEDNAME-VSPTR  POINTER.
** Offset of variable length string
20 MQIMPO-RETURNEDNAME-VSOFFSET PIC S9(9) BINARY.
** CCSID of variable length string
20 MQIMPO-RETURNEDNAME-VSCCSID PIC S9(9) BINARY.
** Property data type as string
15 MQIMPO-TYPESTRING      PIC S9(9) BINARY.
```

## Declaração PL/I para MQIMPO

```
dc1
1 MQIMPO based,
3 StructId          char(4),          /* Structure identifier */
3 Version           fixed bin(31),    /* Structure version number */
3 Options           fixed bin(31),    /* Options that control the
                                     action of MQINQMP */
3 RequestedEncoding fixed bin(31),    /* Requested encoding of
                                     Value */
3 RequestedCCSID    fixed bin(31),    /* Requested character set
                                     identifier of Value */
3 ReturnedEncoding  fixed bin(31),    /* Returned encoding of
                                     Value */
3 ReturnedCCSID     fixed bin(31),    /* Returned character set
                                     identifier of Value */
3 Reserved1         fixed bin(31),    /* Reserved field */
3 ReturnedName,     /* Returned property name */
5 ReturnedName_VSPtr pointer,         /* Address of returned
                                     name */
5 5 ReturnedName_VSOFFSET fixed bin(31), /* Offset of returned
                                     name */
5 5 ReturnedName_VSCCSID fixed bin(31), /* CCSID of returned
                                     name */
3 TypeString        char(8);          /* Property data type as
                                     string */
```

## Declaração High Level Assembler para MQIMPO

```
MQIMPO          DSECT
MQIMPO_STRUCID  DS   CL4  Structure identifier
MQIMPO_VERSION  DS   F    Structure version number
MQIMPO_OPTIONS  DS   F    Options that control the
*               action of MQINQMP
MQIMPO_REQUESTEDENCODING DS F    Requested encoding of VALUE
MQIMPO_REQUESTEDCCSID   DS F    Requested character set
*               identifier of VALUE
MQIMPO_RETURNEDENCODING DS F    Returned encoding of VALUE
MQIMPO_RETURNEDCCSID    DS F    Returned character set
*               identifier of VALUE
MQIMPO_RESERVED1       DS   F    Reserved field
MQIMPO_RETURNEDNAME     DS   0F  Force fullword alignment
MQIMPO_RETURNEDNAME_VSPTR DS   F    Address of returned name
MQIMPO_RETURNEDNAME_VSOFFSET DS F    Offset of returned name
MQIMPO_RETURNEDNAME_VSLENGTH DS F    Length of returned name
MQIMPO_RETURNEDNAME_VSCCSID DS F    CCSID of returned name
MQIMPO_RETURNEDNAME_LENGTH EQU  *-MQIMPO_RETURNEDNAME
MQIMPO_RETURNEDNAME_AREA DS   CL(MQIMPO_RETURNEDNAME_LENGTH)
*
MQIMPO_TYPESTRING      DS   CL8  Property data type as string
MQIMPO_LENGTH          EQU  *-MQIMPO
MQIMPO_AREA            DS   CL(MQIMPO_LENGTH)
```

### **StrucId (MQCHAR4) para MQIMPO**

Esse é o identificador de estrutura da estrutura de opções de propriedades de mensagens de consulta.. Ele é sempre um campo de entrada Seu valor é MQIMPO\_STRUC\_ID.

O valor deve ser:.

### **MQIMPO\_STRUC\_ID**

Identificador para a estrutura de opções de propriedades de mensagens de consulta.

Para a linguagem de programação C, a constante MQIMPO\_STRUC\_ID\_ARRAY também é definida. Ele tem o mesmo valor que MQIMPO\_STRUC\_ID, mas é uma matriz de caracteres em vez de uma sequência.

### **Versão (MQLONG) para MQIMPO**

Estrutura de opções de propriedades de mensagem de consulta-campo Versão..

Este é o número da versão da estrutura O valor deve ser:.



## **MQIMPO\_VERSION\_1**

Número da versão para a estrutura de opções de propriedades de mensagem de consulta

A constante a seguir especifica o número da versão atual:

## **MQIMPO\_CURRENT\_VERSION**

Versão atual da estrutura de opções de propriedades de mensagem de consulta

Isto é sempre um campo de entrada. O valor inicial desse campo é `MQIMPO_VERSION_1`.

## **Opções (MQLONG) para MQIMPO**

Estrutura de opções de propriedade de mensagem de consulta-campo Opções..

As opções a seguir controlam a ação de `MQINQMP`. É possível especificar uma ou mais dessas opções. Para especificar mais de uma opção, inclua os valores juntos (não inclua a mesma constante mais de uma vez) ou combine os valores usando a operação OR bit a bit (se a linguagem de programação suportar operações de bit).

Combinações de opções inválidas são observadas; todas as outras combinações são válidas.

**Opções de dados de valor:** As opções a seguir estão relacionadas ao processamento dos dados de valor quando a propriedade é recuperada da mensagem.

## **MQIMPO\_CONVERT\_VALUE**

Essa opção solicita que o valor da propriedade seja convertido para estar em conformidade com os valores *RequestedCCSID* e *RequestedEncoding* especificados antes da chamada `MQINQMP` retornar o valor da propriedade na área *Value*.

- Se a conversão for bem-sucedida, os campos *ReturnedCCSID* e *ReturnedEncoding* serão configurados como *RequestedCCSID* e *RequestedEncoding* no retorno da chamada `MQINQMP`.
- Se a conversão falhar, mas a chamada `MQINQMP` for concluída sem erro, o valor da propriedade será retornado sem conversão.

Se a propriedade for uma sequência, os campos *ReturnedCCSID* e *ReturnedEncoding* serão configurados para o conjunto de caracteres e a codificação da sequência não convertida.

O código de conclusão é `MQCC_WARNING` neste caso, com código de razão `MQRC_PROP_VALUE_NOT_CONVERTED`. O cursor da propriedade é avançado para a propriedade retornada.

Se o valor da propriedade se expandir durante a conversão e exceder o tamanho do parâmetro **Value**, o valor será retornado não convertido, com o código de conclusão `MQCC_FAILED`; o código de razão é configurado como `MQRC_PROPERTY_VALUE_TOO_BIG`.

O parâmetro **DataLength** da chamada `MQINQMP` retorna o comprimento para o qual o valor da propriedade teria sido convertido, para permitir que o aplicativo determine o tamanho do buffer necessário para acomodar o valor da propriedade convertido. O cursor da propriedade está inalterado.

Essa opção também solicita que:

- Se o nome da propriedade contiver um curinga e
- O campo *ReturnedName* é inicializado com um endereço ou deslocamento para o nome retornado, então o nome retornado será convertido para estar em conformidade com os valores *RequestedCCSID* e *RequestedEncoding*.
- Se a conversão for bem-sucedida, o campo *VSCCSID* do *ReturnedName* e a codificação do nome retornado serão configurados para o valor de entrada de *RequestedCCSID* e *RequestedEncoding*.
- Se a conversão falhar, mas a chamada `MQINQMP` for concluída sem erro ou aviso, o nome retornado não será convertido. O código de conclusão é `MQCC_WARNING` neste caso, com código de razão `MQRC_PROP_NAME_NOT_CONVERTED`.

O cursor da propriedade é avançado para a propriedade retornada  
 MQRC\_PROP\_VALUE\_NOT\_CONVERTED será retornado se o valor e o nome não forem convertidos..

Se o nome retornado for expandido durante a conversão e exceder o tamanho do campo *VSBuFSIZE* do *RequestedName*, a sequência retornada não será convertida, com o código de conclusão MQCC\_FAILED e o código de razão será configurado como MQRC\_PROPERTY\_NAME\_TOO\_BIG..

O campo *VSLength* da estrutura MQCHARV retorna o comprimento para o qual o valor da propriedade teria sido convertido, para permitir que o aplicativo determine o tamanho do buffer necessário para acomodar o valor da propriedade convertido. O cursor da propriedade está inalterado

### MQIMPO\_CONVERT\_TYPE

Esta opção solicita que o valor da propriedade seja convertido de seu tipo de dados atual para o tipo de dados especificado no parâmetro **Type** da chamada MQINQMP

- Se a conversão for bem-sucedida, o parâmetro **Type** permanecerá inalterado no retorno da chamada MQINQMP
- Se a conversão falhar, mas a chamada MQINQMP for concluída sem erro, a chamada falhará com a razão MQRC\_PROP\_CONV\_NOT\_SUPPORTED. O cursor da propriedade está inalterado

Se a conversão do tipo de dados fizer com que o valor seja expandido durante a conversão e o valor convertido exceder o tamanho do parâmetro **Value**, o valor será retornado não convertido, com o código de conclusão MQCC\_FAILED e o código de razão será configurado como MQRC\_PROPERTY\_TOO\_BIG.

O parâmetro **DataLength** da chamada MQINQMP retorna o comprimento para o qual o valor da propriedade teria sido convertido, para permitir que o aplicativo determine o tamanho do buffer necessário para acomodar o valor da propriedade convertido. O cursor da propriedade está inalterado

Se o valor do parâmetro **Type** da chamada MQINQMP não for válido, a chamada falha com a razão MQRC\_PROPERTY\_TYPE\_ERROR.

Se a conversão de tipo de dados solicitada não for suportada, a chamada falhará com a razão MQRC\_PROP\_CONV\_NOT\_SUPPORTED. As seguintes conversões de tipo de dados são suportadas:

Tabela 499. Conversões de tipo de dados suportadas	
Tipo de Dados da Propriedade	Tipos de dados de destino suportados
MQTYPE_BOOLEAN	MQTYPE_STRING, MQTYPE_INT8, MQTYPE_INT16, MQTYPE_INT32, MQTYPE_INT64
MQTYPE_BYTE_STRING	MQTYPE_STRING
MQTYPE_INT8	MQTYPE_STRING, MQTYPE_INT16; MQTYPE_INT32, MQTYPE_INT64
MQTYPE_INT16	MQTYPE_STRING, MQTYPE_INT32, MQTYPE_INT64 .
MQTYPE_INT32	MQTYPE_STRING, MQTYPE_INT64
MQTYPE_INT64	MQTYPE_STRING
MQTYPE_FLOAT32	MQTYPE_STRING, MQTYPE_FLOAT64
MQTYPE_FLOAT64	MQTYPE_STRING
MQTYPE_STRING	MQTYPE_BOOLEAN, MQTYPE_INT8, MQTYPE_INT16, MQTYPE_INT32, MQTYPE_INT64, MQTYPE_FLOAT32, MQTYPE_FLOAT64 .
MQTYPE_NULL	Nenhum

As regras gerais que regem as conversões suportadas são as seguintes:

- Valores de propriedade numérica podem ser convertidos de um tipo de dados para outro, desde que nenhum dado seja perdido durante a conversão.

Por exemplo, o valor de uma propriedade com o tipo de dados MQTYPE\_INT32 pode ser convertido em um valor com o tipo de dados MQTYPE\_INT64, mas não pode ser convertido em um valor com o tipo de dados MQTYPE\_INT16.

- Um valor de propriedade de qualquer tipo de dados pode ser convertido em uma sequência.
- Um valor de propriedade de sequência pode ser convertido para qualquer outro tipo de dados, desde que a sequência seja formatada corretamente para a conversão. Se um aplicativo tentar converter um valor de propriedade de cadeia que não esteja formatado corretamente, IBM MQ retornará o código de razão MQRC\_PROP\_NUMBER\_FORMAT\_ERROR.
- Se um aplicativo tentar uma conversão que não seja suportada, o IBM MQ retornará o código de razão MQRC\_PROP\_CONV\_NOT\_SUPPORTED.

As regras específicas para converter um valor de propriedade de um tipo de dados para outro são as seguintes:

- Ao converter um valor da propriedade MQTYPE\_BOOLEAN em uma sequência, o valor TRUE é convertido na sequência "TRUE" e o valor false é convertido na sequência "FALSE".
- Ao converter um valor da propriedade MQTYPE\_BOOLEAN em um tipo de dado numérico, o valor TRUE é convertido em um e o valor FALSE é convertido em zero..
- Ao converter um valor de propriedade de cadeia para um valor MQTYPE\_BOOLEAN, a cadeia "TRUE", ou "1", é convertida para TRUE e a cadeia "FALSE", ou "0", é convertida para FALSE.

Observe que os termos "TRUE" e "FALSE" não fazem distinção entre maiúsculas e minúsculas.

Nenhuma outra sequência pode ser convertida; IBM MQ retorna o código de razão MQRC\_PROP\_NUMBER\_FORMAT\_ERROR.

- Ao converter um valor de propriedade de sequência para um valor com o tipo de dado MQTYPE\_INT8, MQTYPE\_INT16, MQTYPE\_INT32 ou MQTYPE\_INT64, a sequência deve ter o formato a seguir:

```
[blanks][sign]digits
```

Os significados dos componentes da sequência são os seguintes:

**blanks**

Caracteres em branco iniciais opcionais

**sign**

Um caractere de sinal de mais (+) ou de sinal de menos (-) opcional.

**digits**

Uma sequência contígua de caracteres de dígito (0-9). Pelo menos um caractere de dígito deve estar presente.

Após a sequência de caracteres de dígito, a sequência pode conter outros caracteres que não são caracteres de dígito, mas a conversão para assim que o primeiro desses caracteres for atingido. A sequência é assumida para representar um número inteiro decimal.

IBM MQ retornará o código de razão MQRC\_PROP\_NUMBER\_FORMAT\_ERROR se a sequência não estiver formatada corretamente

- Ao converter um valor da propriedade de cadeia para um valor com o tipo de dados MQTYPE\_FLOAT32 ou MQTYPE\_FLOAT64, a cadeia deve ter o seguinte formato:

```
[blanks][sign]digits[.digits][e_char[e_sign]e_digits]
```

Os significados dos componentes da sequência são os seguintes:

**blanks**

Caracteres em branco iniciais opcionais

**sign**

Um caractere de sinal de mais (+) ou de sinal de menos (-) opcional.

**digits**

Uma sequência contígua de caracteres de dígito (0-9). Pelo menos um caractere de dígito deve estar presente.

**e\_char**

Um caractere de expoente, que é "E" ou "e".

**e\_sign**

Um caractere de sinal de mais (+) ou sinal de menos (-) opcional para o expoente.

**e\_digits**

Uma sequência contígua de caracteres de dígito (0-9) para o expoente. Pelo menos um caractere de dígito deve estar presente se a sequência contiver um caractere expoente.

Após a sequência de caracteres de dígito ou os caracteres opcionais que representam um expoente, a sequência pode conter outros caracteres que não são caracteres de dígito, mas a conversão para assim que o primeiro desses caracteres for atingido. Supõe-se que a sequência represente um número de vírgula flutuante decimal com um expoente que é uma potência de 10.

IBM MQ retornará o código de razão MQRCPROP\_NUMBER\_FORMAT\_ERROR se a sequência não estiver formatada corretamente

- Ao converter um valor de propriedade numérica em uma sequência, o valor é convertido para a representação de sequência do valor como um número decimal, não a sequência que contém o caractere ASCII para esse valor... Por exemplo, o número inteiro 65 é convertido para a sequência "65", não a sequência "A".
- Ao converter um valor de propriedade de sequência de bytes em uma sequência, cada byte é convertido para os dois caracteres hexadecimais que representam o byte... Por exemplo, a matriz de bytes {0xF1, 0x12, 0x00, 0xFF} é convertida na sequência "F11200FF".

**MQIMPO\_QUERY\_LENGTH**

Consulte o tipo e comprimento do valor da propriedade. O comprimento é retornado no parâmetro **DataLength** da chamada MQINQMP.. O valor da propriedade não é retornado.

Se um buffer **ReturnedName** for especificado, o campo *VSLength* da estrutura MQCHARV será preenchido com o comprimento do nome da propriedade.. O nome da propriedade não é retornado..

**Opções de iteração:** As opções a seguir estão relacionadas à iteração sobre propriedades, usando um nome com um caractere curinga

**MQIMPO\_INQ\_FIRST**

Consulte a primeira propriedade correspondente ao nome especificado. Após essa chamada, um cursor é estabelecido na propriedade retornada.

Esse é o valor-padrão.

A opção MQIMPO\_INQ\_PROP\_UNDER\_CURSOR pode ser usada subsequentemente com uma chamada MQINQMP, se necessário, para consultar a mesma propriedade novamente.

Observe que há apenas um cursor de propriedade; portanto, se o nome da propriedade, especificado na chamada MQINQMP, alterar o cursor será reconfigurado

Esta opção não é válida com uma das seguintes opções:

MQIMPO\_INQ\_NEXT

MQIMPO\_INQ\_PROP\_UNDER\_CURSOR

**MQIMPO\_INQ\_NEXT**

Consulta sobre a próxima propriedade que corresponde ao nome especificado, continuando a procura a partir do cursor de propriedade O cursor é avançado para a propriedade retornada.

Se esta for a primeira chamada MQINQMP para o nome especificado, então a primeira propriedade que corresponde ao nome especificado será retornada

A opção MQIMPO\_INQ\_PROP\_UNDER\_CURSOR pode ser usada subsequentemente com uma chamada MQINQMP, se necessário, para consultar a mesma propriedade novamente.

Se a propriedade sob o cursor tiver sido excluída, MQINQMP retornará a próxima propriedade correspondente após aquela que foi excluída

Se for incluída uma propriedade que corresponda ao curinga, enquanto uma iteração estiver em andamento, a propriedade poderá ou não ser retornada durante a conclusão da iteração. A propriedade é retornada quando a iteração é reiniciada usando MQIMPO\_INQ\_FIRST.

Uma propriedade correspondente ao curinga que foi excluído, enquanto a iteração estava em andamento, não é retornada após a sua exclusão

Esta opção não é válida com uma das seguintes opções:

MQIMPO\_INQ\_FIRST  
MQIMPO\_INQ\_PROP\_UNDER\_CURSOR

#### **MQIMPO\_INQ\_PROP\_UNDER\_CURSOR**

Recuperar o valor da propriedade apontada pelo cursor da propriedade. A propriedade apontada pelo cursor da propriedade é aquela que foi consultada pela última vez, usando a opção MQIMPO\_INQ\_FIRST ou MQIMPO\_INQ\_NEXT.

O cursor de propriedade é reconfigurado quando o identificador de mensagens é reutilizado, quando o identificador de mensagens é especificado no campo *MsgHandle* do MQGMO em uma chamada MQGET ou quando o identificador de mensagens é especificado nos campos *OriginalMsgHandle* ou *NewMsgHandle* da estrutura MQPMO em uma chamada MQPUT

Se essa opção for utilizada quando o cursor de propriedade ainda não tiver sido estabelecido, ou se a propriedade apontada pelo cursor de propriedade tiver sido excluída, a chamada falhará com o código de conclusão MQCC\_FAILED e a razão MQRC\_PROPERTY\_NOT\_AVAILABLE

Esta opção não é válida com uma das seguintes opções:

MQIMPO\_INQ\_FIRST  
MQIMPO\_INQ\_NEXT

Se nenhuma das opções descritas anteriormente for necessária, a seguinte opção poderá ser usada:

#### **MQIMPO\_NONE**

Use esse valor para indicar que nenhuma outra opção foi especificada. Todas as opções assumem seus valores-padrão.

MQIMPO\_NONE ajuda a documentação do programa; não se pretende que essa opção seja usada com nenhuma outra, mas como seu valor é zero, tal uso não pode ser detectado.

Isto é sempre um campo de entrada. O valor inicial desse campo é MQIMPO\_INQ\_FIRST.

#### ***RequestedEncoding (MQLONG) para MQIMPO***

Consultar a estrutura de opções da propriedade de mensagem-campo RequestedEncoding ....

Essa é a codificação na qual o valor da propriedade consultada deve ser convertido quando MQIMPO\_CONVERT\_VALUE ou MQIMPO\_CONVERT\_TYPE for especificado.

O valor inicial desse campo é MQENC\_NATIVE.

#### ***RequestedCCSID (MQLONG) para MQIMPO***

Estrutura de opções de propriedade de mensagem de consulta-campo RequestedCCSID

O conjunto de caracteres em que o valor da propriedade consultada deve ser convertido se o valor for uma cadeia de caracteres. Esse também é o conjunto de caracteres no qual o *ReturnedName* deve ser convertido quando MQIMPO\_CONVERT\_VALUE ou MQIMPO\_CONVERT\_TYPE é especificado..

O valor inicial desse campo é MQCCSI\_APPL.

### ***ReturnedEncoding (MQLONG) para MQIMPO***

Estrutura de opções de propriedade de mensagem de consulta-campo ReturnedEncoding ..

Na saída, esta é a codificação do valor retornado.

Se a opção MQIMPO\_CONVERT\_VALUE for especificada e a conversão foi bem-sucedida, o campo *ReturnedEncoding* , no retorno, será o mesmo valor que o valor transmitido.

O valor inicial desse campo é MQENC\_NATIVE.

### ***ReturnedCCSID (MQLONG) para MQIMPO***

Estrutura de opções de propriedades de mensagem de consulta-campo ReturnedCCSID

Na saída, esse é o conjunto de caracteres do valor retornado se o parâmetro **Type** da chamada MQINQMP for MQTYPE\_STRING.

Se a opção MQIMPO\_CONVERT\_VALUE for especificada e a conversão foi bem-sucedida, o campo *ReturnedCCSID* , no retorno, será o mesmo valor que o valor transmitido.

O valor inicial deste campo é zero.

### ***Reserved1 (MQCHAR) para MQIMPO***

Este é um campo reservado O valor inicial deste campo é um caractere em branco (campo de 4 bytes).

### ***ReturnedName (MQCHARV) para MQIMPO***

Estrutura de opções de propriedades de mensagem de consulta-campo ReturnedName

O nome real da propriedade consultada..

Na entrada, um buffer de sequência pode ser transmitido usando o campo *VSPtr* ou *VSOffset* da estrutura MQCHARV O comprimento do buffer de sequência é especificado usando o campo *VSBuFSIZE* da estrutura MQCHARV.

No retorno da chamada MQINQMP, o buffer de sequência é concluído com o nome da propriedade que foi consultada, contanto que o buffer de sequência tenha sido longo o suficiente para conter totalmente o nome. O campo *VSLength* da estrutura MQCHARV é preenchido com o comprimento do nome da propriedade.. O campo *VSCCSID* da estrutura MQCHARV é preenchido para indicar o conjunto de caracteres do nome retornado, se a conversão do nome falhou.

Este é um campo de entrada/saída O valor inicial desse campo é MQCHARV\_DEFAULT.

### ***TypeString (MQCHAR8) para MQIMPO.***

Estrutura de opções de propriedades de mensagem de consulta-campo TypeString

Uma representação em cadeia do tipo de dados da propriedade

Se a propriedade foi especificada em um cabeçalho MQRFH2 e o atributo MQRFH2 dt não for reconhecido, esse campo poderá ser usado para determinar o tipo de dado da propriedade. *TypeString* é retornado no conjunto de caracteres codificados 1208 (UTF-8) e é os primeiros oito bytes do valor do atributo dt da propriedade que falhou ao ser reconhecido

Este é sempre um campo de saída O valor inicial deste campo é a cadeia nula na linguagem de programação C e 8 caracteres em branco em outras linguagens de programação.

## MQMD - descritor de mensagem

A estrutura MQMD contém as informações de controle que acompanham os dados do aplicativo quando uma mensagem viaja entre os aplicativos de envio e recebimento. A estrutura é um parâmetro de entrada / saída nas chamadas MQGET, MQPUT e MQPUT1 ..

## Disponibilidade

Todos os sistemas IBM MQ , além de IBM MQ MQI clients conectados a esses sistemas

## Versão

A versão atual do MQMD é MQMD\_VERSION\_2. Os aplicativos que devem ser móveis entre vários ambientes devem assegurar que a versão necessária do MQMD seja suportada em todos os ambientes em questão. Os campos que existem apenas nas versões mais recentes da estrutura são identificados como tal nas descrições a seguir:

Os arquivos de cabeçalho, COPY e INCLUDE fornecidos para as linguagens de programação suportadas contêm a versão mais recente do MQMD suportada pelo ambiente, mas com o valor inicial do campo *Version* configurado como MQMD\_VERSION\_1. Para usar campos que não estão presentes na estrutura version-1 , o aplicativo deve configurar o campo *Version* para o número da versão necessária.

Uma declaração para a estrutura version-1 está disponível com o nome MQMD1.

## Conjunto de caracteres e codificação

Os dados no MQMD devem estar no conjunto de caracteres e na codificação do gerenciador de filas locais; eles serão fornecidos pelo atributo do gerenciador de filas do **CodedCharSetId** e MQENC\_NATIVE. No entanto, se o aplicativo estiver em execução como um IBM MQ MQI client, a estrutura deverá estar no conjunto de caracteres e na codificação do cliente.

Se os gerenciadores de filas de envio e de recebimento usarem conjuntos de caracteres ou codificações diferentes, os dados no MQMD serão convertidos automaticamente. Não é necessário que o aplicativo converta o MQMD.

## Usando diferentes versões de MQMD

Um MQMD version-2 é equivalente a usar um MQMD version-1 e prefixar os dados da mensagem com uma estrutura MQMDE. No entanto, se todos os campos na estrutura MQMDE tiverem seus valores padrão, MQMDE poderá ser omitido. Um MQMD version-1 mais MQMDE são usados conforme descrito:

- Nas chamadas MQPUT e MQPUT1 , se o aplicativo fornecer um MQMD version-1 , o aplicativo poderá opcionalmente prefixar os dados da mensagem com um MQMDE, configurando o campo *Format* em MQMD para MQFMT\_MD\_EXTENSION para indicar que um MQMDE está presente. Se o aplicativo não fornecer um MQMDE, o gerenciador de filas assume valores padrão para os campos no MQMDE.

**Nota:** Vários dos campos que existem no MQMD version-2 mas não no MQMD version-1 são campos de entrada / saída nas chamadas MQPUT e MQPUT1 . No entanto, o gerenciador de filas não retorna nenhum valor nos campos equivalentes no MQMDE na saída das chamadas MQPUT e MQPUT1 ; se o aplicativo requerer esses valores de saída, ele deverá usar um MQMD version-2 .

- Na chamada MQGET, se o aplicativo fornecer um MQMD version-1 , o gerenciador de filas prefixa a mensagem retornada com um MQMDE, mas somente se um ou mais dos campos no MQMDE tiver um valor não padrão. O campo *Format* no MQMD terá o valor MQFMT\_MD\_EXTENSION para indicar que um MQMDE está presente

Os valores padrão que o gerenciador de filas usa para os campos no MQMDE são iguais aos valores iniciais desses campos, mostrados em [Tabela 503 na página 490](#).

Quando uma mensagem está em uma fila de transmissão, alguns dos campos no MQMD são configurados para valores específicos; consulte [“MQXQH-Cabeçalho da fila de transmissão” na página 641](#) para obter detalhes

## Contexto da mensagem

Determinados campos no MQMD contêm o contexto de mensagem. Há dois tipos de contexto de mensagem: *contexto de identidade* e *contexto de origem*. Tipicamente:

- O contexto de identidade está relacionado ao aplicativo que *originalmente* colocou a mensagem
- O contexto de origem está relacionado ao aplicativo que *mais recentemente* colocou a mensagem.

Esses dois aplicativos podem ser o mesmo aplicativo, mas eles também podem ser aplicativos diferentes (por exemplo, quando uma mensagem é encaminhada de um aplicativo para outro)

Embora a identidade e o contexto de origem geralmente tenham os significados descritos, o conteúdo de ambos os tipos de campos de contextos no MQMD depende das opções MQPMO\_\*\_CONTEXT que são especificadas quando a mensagem é colocada. Como resultado, o contexto de identidade não está necessariamente relacionado ao aplicativo que originalmente colocou a mensagem e o contexto de origem não está necessariamente relacionado ao aplicativo que mais recentemente colocou a mensagem; ele depende do design do conjunto de aplicativos.

O MCA (Message Channel Agent) nunca altera o contexto da mensagem. Os MCAs que recebem mensagens dos gerenciadores de filas remotas usam a opção de contexto MQPMO\_SET\_ALL\_CONTEXT na chamada MQPUT ou MQPUT1. Isso permite que o MCA de recebimento preserve exatamente o contexto da mensagem que viajou com a mensagem do MCA de envio. No entanto, o resultado é que o contexto de origem não está relacionado a nenhum dos MCAs que enviaram e receberam a mensagem. O contexto de origem refere-se a um aplicativo anterior que colocou a mensagem. Se todos os aplicativos intermediários tiverem passado o contexto da mensagem, o contexto de origem se referirá ao próprio aplicativo de origem.

Nas descrições, os campos de contexto são descritos como se fossem usados conforme descrito anteriormente. Para obter mais informações sobre contexto da mensagem, consulte [Contexto da mensagem](#).

## Campos

**Nota:** Na tabela a seguir, os campos são agrupados por uso e não alfabeticamente. Os tópicos filhos seguem a mesma sequência.

Nome e descrição do campo	Nome da constante	Valor inicial (se houver) da constante
<u>StrucId</u> (identificador de estrutura)	MQMD_STRUC_ID	'MD'
<u>Versão</u> (número da versão da estrutura).	MQMD_VERSION_1	1
<u>Relatório</u> (opções para mensagens de relatório)	MQRO_NONE	0
<u>MsgType</u> (tipo de mensagem).	MQMT_DATAGRAM	8
<u>MQMD-Campo de expiração</u> (tempo de vida da mensagem).	MQEI_UNLIMITED	-1
<u>MQMD-Campo de feedback</u> (feedback ou código de razão)	MQFB_NONE	0
<u>Codificação</u> (codificação numérica de dados da mensagem)..	MQENC_NATIVE	Depende do ambiente
<u>CodedCharSetId</u> (identificador do conjunto de caracteres de dados da mensagens)	MQCCSI_Q_MGR	0
<u>Formato</u> (nome do formato dos dados da mensagem)	MQFMT_NONE	Espaços em branco



Tabela 500. Campos em MQMD para MQMD (continuação)

Nome e descrição do campo	Nome da constante	Valor inicial (se houver) da constante
<u>Prioridade</u> (prioridade da mensagem).	MQPRI_PRIORITY_AS_Q_DEF	-1
<u>Persistência</u> (persistência de mensagem).	MQPER_PERSISTENCE_AS_Q_DEF	2
MQMD-campo <u>MsgId</u> (identificador de mensagens).	MQMI_NONE	Nulos
<u>CorrelId</u> (identificador de correlação)	MQCI_NONE	Nulos
<u>BackoutCount</u> (contador de restauração)	Nenhum	0
<u>ReplyToQ</u> (nome da fila de respostas).	Nenhum	Cadeia nula ou espaços em branco
<u>ReplyToQMgr</u> (nome do gerenciador de filas de resposta)	Nenhum	Cadeia nula ou espaços em branco
<u>UserIdentifier</u> (identificador de usuário)	Nenhum	Cadeia nula ou espaços em branco
<u>AccountingToken</u> (token de contabilidade)	MQACT_NONE	Nulos
<u>ApplIdentityDados</u> (dados do aplicativo relacionados à identidade)	Nenhum	Cadeia nula ou espaços em branco
<u>PutApplTipo</u> (tipo de aplicativo que coloca a mensagem)	MQAT_NO_CONTEXT	0
<u>PutApplNome</u> (nome do aplicativo que colocou a mensagem)	Nenhum	Cadeia nula ou espaços em branco
<u>PutDate</u> (data em que a mensagem foi colocada).	Nenhum	Cadeia nula ou espaços em branco
<u>PutTime</u> (horário quando a mensagem foi colocada)	Nenhum	Cadeia nula ou espaços em branco
<u>ApplOriginDados</u> (dados do aplicativo relacionados à origem)	Nenhum	Cadeia nula ou espaços em branco
<b>Nota:</b> Os campos restantes serão ignorados se <i>Version</i> for menor que MQMD_VERSION_2.		
<u>GroupId</u> (identificador de grupo).	MQGI_NONE	Nulos
<u>MsgSeqNúmero</u> (número de sequência da mensagem lógica dentro do grupo)	Nenhum	1
<u>Deslocamento</u> (deslocamento de dados na mensagem física do início da mensagem lógica)	Nenhum	0
MQMD-campo <u>MsgFlags</u> (sinalizações de mensagens)	MQMF_NONE	0
<u>OriginalLength</u> (comprimento da mensagem original)	MQOL_UNDEFINED	-1

Tabela 500. Campos em MQMD para MQMD (continuação)

Nome e descrição do campo	Nome da constante	Valor inicial (se houver) da constante
<b>Notas:</b>		
<p>1. O valor Sequência nula ou espaços em branco indica a sequência nula em C e caracteres em branco em outras linguagens de programação.</p> <p>2. Na linguagem de programação C, a variável macroMQMD_DEFAULT contém os valores listados na tabela. Ele pode ser usado da seguinte maneira para fornecer valores iniciais para os campos na estrutura:</p>		
<pre>MQMD MyMD = {MQMD_DEFAULT};</pre>		

## Declarações de idiomas

### Declaração C para MQMD

```
typedef struct tagMQMD MQMD;
struct tagMQMD {
    MQCHAR4   StrucId;           /* Structure identifier */
    MQLONG    Version;          /* Structure version number */
    MQLONG    Report;           /* Options for report messages */
    MQLONG    MsgType;          /* Message type */
    MQLONG    Expiry;           /* Message lifetime */
    MQLONG    Feedback;         /* Feedback or reason code */
    MQLONG    Encoding;         /* Numeric encoding of message data */
    MQLONG    CodedCharSetId;   /* Character set identifier of message
                                data */

    MQCHAR8   Format;           /* Format name of message data */
    MQLONG    Priority;          /* Message priority */
    MQLONG    Persistence;      /* Message persistence */
    MQBYTE24  MsgId;            /* Message identifier */
    MQBYTE24  CorrelId;         /* Correlation identifier */
    MQLONG    BackoutCount;     /* Backout counter */
    MQCHAR48  ReplyToQ;         /* Name of reply queue */
    MQCHAR48  ReplyToQMGr;      /* Name of reply queue manager */
    MQCHAR12  UserIdentifier;    /* User identifier */
    MQBYTE32  AccountingToken;  /* Accounting token */
    MQCHAR32  ApplIdentityData; /* Application data relating to
                                identity */

    MQLONG    PutApplType;      /* Type of application that put the
                                message */
    MQCHAR28  PutApplName;      /* Name of application that put the
                                message */

    MQCHAR8   PutDate;          /* Date when message was put */
    MQCHAR8   PutTime;          /* Time when message was put */
    MQCHAR4   ApplOriginData;   /* Application data relating to origin */
    MQBYTE24  GroupId;          /* Group identifier */
    MQLONG    MsgSeqNumber;     /* Sequence number of logical message
                                within group */

    MQLONG    Offset;           /* Offset of data in physical message
                                from start of logical message */

    MQLONG    MsgFlags;         /* Message flags */
    MQLONG    OriginalLength;   /* Length of original message */
};
```

### Declaração COBOL para MQMD

```
** MQMD structure
10 MQMD.
** Structure identifier
15 MQMD-STRUCID PIC X(4).
** Structure version number
15 MQMD-VERSION PIC S9(9) BINARY.
** Options for report messages
15 MQMD-REPORT PIC S9(9) BINARY.
** Message type
```

```

15 MQMD-MSGTYPE          PIC S9(9) BINARY.
** Message lifetime
15 MQMD-EXPIRY          PIC S9(9) BINARY.
** Feedback or reason code
15 MQMD-FEEDBACK        PIC S9(9) BINARY.
** Numeric encoding of message data
15 MQMD-ENCODING        PIC S9(9) BINARY.
** Character set identifier of message data
15 MQMD-CODEDCHARSETID  PIC S9(9) BINARY.
** Format name of message data
15 MQMD-FORMAT          PIC X(8).
** Message priority
15 MQMD-PRIORITY        PIC S9(9) BINARY.
** Message persistence
15 MQMD-PERSISTENCE     PIC S9(9) BINARY.
** Message identifier
15 MQMD-MSGID           PIC X(24).
** Correlation identifier
15 MQMD-CORRELID        PIC X(24).
** Backout counter
15 MQMD-BACKOUTCOUNT   PIC S9(9) BINARY.
** Name of reply queue
15 MQMD-REPLYTOQ        PIC X(48).
** Name of reply queue manager
15 MQMD-REPLYTOQMGR     PIC X(48).
** User identifier
15 MQMD-USERIDENTIFIER  PIC X(12).
** Accounting token
15 MQMD-ACCOUNTINGTOKEN PIC X(32).
** Application data relating to identity
15 MQMD-APPLIDENTITYDATA PIC X(32).
** Type of application that put the message
15 MQMD-PUTAPPLTYPE     PIC S9(9) BINARY.
** Name of application that put the message
15 MQMD-PUTAPPLNAME     PIC X(28).
** Date when message was put
15 MQMD-PUTDATE         PIC X(8).
** Time when message was put
15 MQMD-PUTTIME         PIC X(8).
** Application data relating to origin
15 MQMD-APPLORIGINDATA  PIC X(4).
** Group identifier
15 MQMD-GROUPID         PIC X(24).
** Sequence number of logical message within group
15 MQMD-MSGSEQUENBER    PIC S9(9) BINARY.
** Offset of data in physical message from start of logical message
15 MQMD-OFFSET          PIC S9(9) BINARY.
** Message flags
15 MQMD-MSGFLAGS        PIC S9(9) BINARY.
** Length of original message
15 MQMD-ORIGINALLENGTH  PIC S9(9) BINARY.

```

## Declaração PL/I para MQMD

```

dcl
1 MQMD based,
3 StrucId          char(4),          /* Structure identifier */
3 Version          fixed bin(31),   /* Structure version number */
3 Report           fixed bin(31),   /* Options for report messages */
3 MsgType          fixed bin(31),   /* Message type */
3 Expiry           fixed bin(31),   /* Message lifetime */
3 Feedback         fixed bin(31),   /* Feedback or reason code */
3 Encoding         fixed bin(31),   /* Numeric encoding of message
data */
3 CodedCharSetId   fixed bin(31),   /* Character set identifier of
message data */
3 Format           char(8),          /* Format name of message data */
3 Priority         fixed bin(31),   /* Message priority */
3 Persistence      fixed bin(31),   /* Message persistence */
3 MsgId           char(24),         /* Message identifier */
3 CorrelId        char(24),         /* Correlation identifier */
3 BackoutCount     fixed bin(31),   /* Backout counter */
3 ReplyToQ        char(48),         /* Name of reply queue */
3 ReplyToQMgr     char(48),         /* Name of reply queue manager */
3 UserIdentifier   char(12),        /* User identifier */
3 AccountingToken  char(32),        /* Accounting token */
3 ApplIdentityData char(32),        /* Application data relating to
identity */
3 PutApplType     fixed bin(31),   /* Type of application that put the

```

```

3 PutApplName      char(28),      /* Name of application that put the
message */
3 PutDate          char(8),      /* Date when message was put */
3 PutTime          char(8),      /* Time when message was put */
3 ApplOriginData  char(4),      /* Application data relating to
origin */
3 GroupId          char(24),     /* Group identifier */
3 MsgSeqNumber     fixed bin(31), /* Sequence number of logical
message within group */
3 Offset           fixed bin(31), /* Offset of data in physical
message from start of logical
message */
3 MsgFlags         fixed bin(31), /* Message flags */
3 OriginalLength   fixed bin(31); /* Length of original message */

```

## Declaração do High Level Assembler para MQMD

```

MQMD                DSECT
MQMD_STRUCID        DS    CL4    Structure identifier
MQMD_VERSION        DS    F      Structure version number
MQMD_REPORT         DS    F      Options for report messages
MQMD_MSGTYPE        DS    F      Message type
MQMD_EXPIRY         DS    F      Message lifetime
MQMD_FEEDBACK       DS    F      Feedback or reason code
MQMD_ENCODING       DS    F      Numeric encoding of message data
MQMD_CODEDCHARSETID DS    F      Character set identifier of message
*
MQMD_FORMAT         DS    CL8    Format name of message data
MQMD_PRIORITY       DS    F      Message priority
MQMD_PERSISTENCE    DS    F      Message persistence
MQMD_MSGID          DS    XL24   Message identifier
MQMD_CORRELID       DS    XL24   Correlation identifier
MQMD_BACKOUTCOUNT DS    F      Backout counter
MQMD_REPLYTOQ       DS    CL48   Name of reply queue
MQMD_REPLYTOQMGR    DS    CL48   Name of reply queue manager
MQMD_USERIDENTIFIER DS    CL12   User identifier
MQMD_ACCOUNTINGTOKEN DS    XL32  Accounting token
MQMD_APPLIDENTITYDATA DS    CL32  Application data relating to identity
MQMD_PUTAPPLTYPE    DS    F      Type of application that put the
*
MQMD_PUTAPPLNAME    DS    CL28   Name of application that put the
*
MQMD_PUTDATE        DS    CL8    Date when message was put
MQMD_PUTTIME        DS    CL8    Time when message was put
MQMD_APPLORIGINDATA DS    CL4    Application data relating to origin
MQMD_GROUPID        DS    XL24   Group identifier
MQMD_MSGSEQNUMBER   DS    F      Sequence number of logical message
*
MQMD_OFFSET         DS    F      Offset of data in physical message
*
MQMD_MSGFLAGS       DS    F      Message flags
MQMD_ORIGINALLENGTH DS    F      Length of original message
*
MQMD_LENGTH         EQU    *-MQMD
ORG    MQMD
MQMD_AREA           DS    CL(MQMD_LENGTH)

```

## Declaração do Visual Basic para MQMD

```

Type MQMD
  StrucId      As String*4 'Structure identifier'
  Version      As Long     'Structure version number'
  Report       As Long     'Options for report messages'
  MsgType      As Long     'Message type'
  Expiry       As Long     'Message lifetime'
  Feedback     As Long     'Feedback or reason code'
  Encoding     As Long     'Numeric encoding of message data'
  CodedCharSetId As Long   'Character set identifier of message'
  'data'
  Format       As String*8  'Format name of message data'
  Priority     As Long     'Message priority'
  Persistence  As Long     'Message persistence'
  MsgId       As MQBYTE24  'Message identifier'
  CorrelId    As MQBYTE24  'Correlation identifier'
  BackoutCount As Long     'Backout counter'
  ReplyToQ    As String*48 'Name of reply queue'

```

ReplyToQMgr	As String*48	'Name of reply queue manager'
UserIdentifier	As String*12	'User identifier'
AccountingToken	As MQBYTE32	'Accounting token'
ApplIdentityData	As String*32	'Application data relating to identity'
PutApplType	As Long	'Type of application that put the 'message'
PutApplName	As String*28	'Name of application that put the 'message'
PutDate	As String*8	'Date when message was put'
PutTime	As String*8	'Time when message was put'
ApplOriginData	As String*4	'Application data relating to origin'
GroupId	As MQBYTE24	'Group identifier'
MsgSeqNumber	As Long	'Sequence number of logical message' 'within group'
Offset	As Long	'Offset of data in physical message' 'from start of logical message'
MsgFlags	As Long	'Message flags'
OriginalLength	As Long	'Length of original message'
End Type		

### **StrucId (MQCHAR4) para MQMD**

Este é o identificador da estrutura do descritor de mensagens. Ele é sempre um campo de entrada Seu valor é MQMD\_STRUC\_ID.

O valor deve ser.:

#### **MQMD\_STRUC\_ID**

Identificador da estrutura do descritor de mensagens.

Para a linguagem de programação C, a constante MQMD\_STRUC\_ID\_ARRAY também é definida. Ele tem o mesmo valor que MQMD\_STRUC\_ID, mas é uma matriz de caracteres em vez de uma sequência.

### **Versão (MQLONG) para MQMD**

Este é o número da versão da estrutura e deve ser um dos seguintes.:

#### **MQMD\_VERSION\_1**

Estrutura do descritor de mensagens Version-1

Esta versão é suportada em todos os ambientes

#### **MQMD\_VERSION\_2**

Estrutura do descritor de mensagens Version-2 .

Essa versão é suportada em todos os ambientes do IBM MQ V6.0 e posterior, mais IBM MQ MQI clients conectados a esses sistemas.

**Nota:** Quando um MQMD version-2 é usado, o gerenciador de filas executa verificações adicionais em quaisquer estruturas de cabeçalho do MQ que possam estar presentes no início dos dados da mensagem do aplicativo; para obter detalhes adicionais, consulte as notas de uso para a chamada MQPUT.

Os campos que existem apenas na versão mais recente da estrutura são identificados como tal nas descrições dos campos A constante a seguir especifica o número da versão atual:

#### **MQMD\_CURRENT\_VERSION**

A versão atual da estrutura do descritor de mensagem

Isto é sempre um campo de entrada. O valor inicial desse campo é MQMD\_VERSION\_1.

### **Relatório (MQLONG) para MQMD**

Uma mensagem de relatório é uma mensagem sobre outra mensagem, usada para informar um aplicativo sobre eventos esperados ou inesperados relacionados à mensagem original. O campo *Report* permite que o aplicativo que está enviando a mensagem original especifique quais mensagens de relatório são necessárias, se os dados da mensagem do aplicativo devem ser incluídos neles e também (para relatórios e respostas) como a mensagem e os identificadores de correlação no relatório ou mensagem de resposta

devem ser configurados. Qualquer ou todos (ou nenhum) dos seguintes tipos de mensagem de relatório pode ser solicitado:

- Exceção
- Expiração
- Confirmar na chegada (COA)
- Confirmar na entrega (COD)
- notificação de ação positiva (PAN)
- notificação de ação negativa (NAN)

É possível especificar uma ou mais dessas opções Para especificar mais de uma opção, inclua os valores juntos (não inclua a mesma constante mais de uma vez) ou combine os valores usando a operação OR bit a bit (se a linguagem de programação suportar operações de bit).

O aplicativo que recebe a mensagem de relatório pode determinar o motivo pelo qual o relatório foi gerado examinando o campo *Feedback* no MQMD; consulte o campo *Feedback* para obter mais detalhes..

O uso de opções de relatório ao colocar uma mensagem em um tópico pode fazer com que zero, uma ou muitas mensagens de relatório sejam geradas e enviadas para o aplicativo. Isso porque a mensagem de publicação pode ser enviada para zero, um ou muitos aplicativos de assinatura.

**Opções de exceção:** Especifique uma das opções listadas para solicitar uma mensagem de relatório de exceção.

#### **MQRO\_EXCEPTION**

Um agente do canal de mensagens gera esse tipo de relatório quando uma mensagem é enviada para outro gerenciador de filas e a mensagem não pode ser entregue para a fila de destino especificada... Por exemplo, a fila de destino ou uma fila de transmissão intermediária pode estar cheia, ou a mensagem pode ser muito grande para a fila

A geração da mensagem de relatório de exceção depende da persistência da mensagem original e da velocidade do canal de mensagens (normal ou rápido) por meio do qual a mensagem original viaja:

- Para todas as mensagens persistentes e para mensagens não persistentes que viajam por canais de mensagens normais, o relatório de exceção será gerado somente se a ação especificada pelo aplicativo de envio para a condição de erro puder ser concluída com êxito. O aplicativo de envio pode especificar uma das seguintes ações para controlar a disposição da mensagem original quando surgir a condição de erro:
  - MQRO\_DEAD\_LETTER\_Q (isto coloca a mensagem original na fila de mensagens não entregues).
  - MQRO\_DISCARD\_MSG (isso descarta a mensagem original).

Se a ação especificada pelo aplicativo de envio não puder ser concluída com êxito, a mensagem original será deixada na fila de transmissão e nenhuma mensagem de relatório de exceção será gerada..

- Para mensagens não persistentes que viajam por canais de mensagens rápidas, a mensagem original é removida da fila de transmissão e o relatório de exceções gerado *mesmo se* a ação especificada para a condição de erro não puder ser concluída com êxito. Por exemplo, se MQRO\_DEAD\_LETTER\_Q for especificado, mas a mensagem original não puder ser colocada na fila de mensagens não entregues porque essa fila está cheia, a mensagem de relatório de exceção será gerada e a mensagem original descartada.

Para obter mais informações sobre canais de mensagens normais e rápidos, consulte [Velocidade da mensagem não persistente \(NPMSPEED\)](#).

Um relatório de exceção não será gerado se o aplicativo que colocou a mensagem original puder ser notificado de forma síncrona sobre o problema por meio do código de razão retornado pela chamada MQPUT ou MQPUT1 .

Os aplicativos também podem enviar relatórios de exceções para indicar que uma mensagem não pode ser processada (por exemplo, porque é uma transação de débito que faria com que a conta excedesse seu limite de crédito).

Os dados da mensagem original não são incluídos com a mensagem de relatório.

Não especifique mais de um MQRO\_EXCEPTION, MQRO\_EXCEPTION\_WITH\_DATA e MQRO\_EXCEPTION\_WITH\_FULL\_DATA.

#### **MQRO\_EXCEPTION\_WITH\_DATA**

Isso é o mesmo que MQRO\_EXCEPTION, exceto que os primeiros 100 bytes dos dados da mensagem do aplicativo da mensagem original são incluídos na mensagem de relatório. Se a mensagem original contiver uma ou mais estruturas de cabeçalho do MQ, elas serão incluídas na mensagem de relatório, além dos 100 bytes de dados do aplicativo

Não especifique mais de um MQRO\_EXCEPTION, MQRO\_EXCEPTION\_WITH\_DATA e MQRO\_EXCEPTION\_WITH\_FULL\_DATA.

#### **MQRO\_EXCEPTION\_WITH\_FULL\_DATA**

Relatórios de exceção com dados completos necessários.

Isso é o mesmo que MQRO\_EXCEPTION, exceto que todos os dados da mensagem do aplicativo da mensagem original estão incluídos na mensagem de relatório

Não especifique mais de um MQRO\_EXCEPTION, MQRO\_EXCEPTION\_WITH\_DATA e MQRO\_EXCEPTION\_WITH\_FULL\_DATA.

**Opções de expiração:** Especifique uma das opções listadas para solicitar uma mensagem de relatório de expiração.

#### **MQRO\_EXPIRATION**

Esse tipo de relatório será gerado pelo gerenciador de filas se a mensagem for descartada antes da entrega para um aplicativo porque seu tempo de expiração passou (consulte o campo *Expiry*). Se essa opção não for configurada, nenhuma mensagem de relatório será gerada se uma mensagem for descartada por essa razão (mesmo se você especificar uma das opções de MQRO\_EXCEPTION\_\*).

Os dados da mensagem original não são incluídos com a mensagem de relatório.

Não especifique mais de um MQRO\_EXPIRATION, MQRO\_EXPIRATION\_WITH\_DATA e MQRO\_EXPIRATION\_WITH\_FULL\_DATA.

#### **MQRO\_EXPIRATION\_WITH\_DATA**

Isso é o mesmo que MQRO\_EXPIRATION, exceto que os primeiros 100 bytes dos dados da mensagem do aplicativo da mensagem original são incluídos na mensagem de relatório. Se a mensagem original contiver uma ou mais estruturas de cabeçalho do MQ, elas serão incluídas na mensagem de relatório, além dos 100 bytes de dados do aplicativo

Não especifique mais de um MQRO\_EXPIRATION, MQRO\_EXPIRATION\_WITH\_DATA e MQRO\_EXPIRATION\_WITH\_FULL\_DATA.

#### **MQRO\_EXPIRATION\_WITH\_FULL\_DATA**

Isso é o mesmo que MQRO\_EXPIRATION, exceto que todos os dados da mensagem do aplicativo da mensagem original estão incluídos na mensagem de relatório

Não especifique mais de um MQRO\_EXPIRATION, MQRO\_EXPIRATION\_WITH\_DATA e MQRO\_EXPIRATION\_WITH\_FULL\_DATA.

**Confirmar opções de chegada:** Especifique uma das opções listadas para solicitar uma mensagem de relatório de confirmação de chegada.

#### **MQRO\_COA**

Esse tipo de relatório é gerado por um gerenciador de filas que possui a fila de destino quando a mensagem é colocada na fila de destino. Os dados da mensagem original não são incluídos com a mensagem de relatório.

Se a mensagem for colocada como parte de uma unidade de trabalho, e a fila de destino for uma fila local, a mensagem de relatório do COA gerada pelo gerenciador de filas poderá ser recuperada somente se a unidade de trabalho for confirmada

Um relatório de COA não será gerado se o campo *Format* no descritor de mensagens for MQFMT\_XMIT\_Q\_HEADER ou MQFMT\_DEAD\_LETTER\_HEADER Isso evita que um relatório COA seja gerado se a mensagem for colocada em uma fila de transmissão ou não for entregue e colocada em uma fila de mensagens não entregues.

No caso de uma fila de ponte IMS , o relatório COA é gerado quando a mensagem atinge a fila IMS (confirmação recebida de IMS ) e não quando a mensagem é colocado na fila de ponte do MQ Isso significa que se IMS não estiver ativo, nenhum relatório COA será gerado até que IMS seja iniciado e uma mensagem seja enfileirada na fila IMS .

O usuário que executa um programa que coloca uma mensagem com MQMD.Report= MQRO\_COA deve ter autoridade + passid na fila de resposta. Se o usuário não tiver a autoridade + passid, a mensagem de relatório COA não atingirá a fila de respostas É feita uma tentativa de colocar a mensagem de relatório na fila de mensagens não entregues.

Não especifique mais de um MQRO\_COA, MQRO\_COA\_WITH\_DATA e MQRO\_COA\_WITH\_FULL\_DATA.

#### **MQRO\_COA\_WITH\_DATA**

Este é o mesmo que MQRO\_COA, exceto que os primeiros 100 bytes dos dados da mensagem do aplicativo da mensagem original são incluídos na mensagem de relatório Se a mensagem original contiver uma ou mais estruturas de cabeçalho do MQ , elas serão incluídas na mensagem de relatório, além dos 100 bytes de dados do aplicativo

Não especifique mais de um MQRO\_COA, MQRO\_COA\_WITH\_DATA e MQRO\_COA\_WITH\_FULL\_DATA.

#### **MQRO\_COA\_WITH\_FULL\_DATA**

Isso é o mesmo que MQRO\_COA, exceto que todos os dados da mensagem do aplicativo da mensagem original estão incluídos na mensagem de relatório

Não especifique mais de um MQRO\_COA, MQRO\_COA\_WITH\_DATA e MQRO\_COA\_WITH\_FULL\_DATA.

**Confirmar opções de entrega:** Especifique uma das opções listadas para solicitar uma mensagem de relatório de confirmação de entrega.

#### **MQRO\_COD**

Esse tipo de relatório é gerado por um gerenciador de filas quando um aplicativo recupera a mensagem da fila de destino de uma maneira que exclui a mensagem da fila. Os dados da mensagem original não são incluídos com a mensagem de relatório.

Se a mensagem for recuperada como parte de uma unidade de serviço, a mensagem de relatório será gerada dentro da mesma unidade de trabalho, de modo que o relatório não estará disponível até que a unidade de trabalho seja confirmada Se a unidade de trabalho for restaurada, o relatório não será enviado..

Um relatório COD nem sempre será gerado se uma mensagem for recuperada com a opção MQGMO\_MARK\_SKIP\_BACKOUT Se a unidade de trabalho primária for restaurada, mas a unidade de trabalho secundária for confirmada, a mensagem será removida da fila, mas um relatório de COD não será gerado

Um relatório COD não será gerado se o campo *Format* no descritor de mensagem for MQFMT\_DEAD\_LETTER\_HEADER.. Isso impede que um relatório COD seja gerado se a mensagem não puder ser entregue e colocada em uma fila de mensagens não entregues.

MQRO\_COD não será válido se a fila de destino for uma fila XCF..

Não especifique mais de um MQRO\_COD, MQRO\_COD\_WITH\_DATA e MQRO\_COD\_WITH\_FULL\_DATA.

#### **MQRO\_COD\_WITH\_DATA**

Isso é o mesmo que MQRO\_COD, exceto que os primeiros 100 bytes dos dados da mensagem do aplicativo da mensagem original são incluídos na mensagem de relatório Se a mensagem original contiver uma ou mais estruturas de cabeçalho do MQ , elas serão incluídas na mensagem de relatório, além dos 100 bytes de dados do aplicativo



Se MQGMO\_ACCEPT\_TRUNCATED\_MSG for especificado na chamada MQGET para a mensagem original e a mensagem recuperada for truncada, a quantia de dados da mensagem do aplicativo colocada na mensagem de relatório dependerá do ambiente:

- Em z/OS, é o mínimo de:
  - O comprimento da mensagem original
  - O comprimento do buffer utilizado para recuperar a mensagem
  - 100 bytes.
- Em outros ambientes, é o mínimo de:
  - O comprimento da mensagem original
  - 100 bytes.

MQRO\_COD\_WITH\_DATA não será válido se a fila de destino for uma fila XCF..

Não especifique mais de um MQRO\_COD, MQRO\_COD\_WITH\_DATA e MQRO\_COD\_WITH\_FULL\_DATA.

### **MQRO\_COD\_WITH\_FULL\_DATA**

Isso é o mesmo que MQRO\_COD, exceto que todos os dados da mensagem do aplicativo da mensagem original são incluídos na mensagem de relatório

MQRO\_COD\_WITH\_FULL\_DATA não será válido se a fila de destino for uma fila do XCF

Não especifique mais de um MQRO\_COD, MQRO\_COD\_WITH\_DATA e MQRO\_COD\_WITH\_FULL\_DATA.

**Opções de notificação de ação:** Especifique uma ou ambas as opções listadas para solicitar que o aplicativo de recebimento envie uma mensagem de relatório de ação positiva ou negativa.

### **MQRO\_PAN**

Esse tipo de relatório é gerado pelo aplicativo que recupera a mensagem e age sobre ela.. Ele indica que a ação solicitada na mensagem foi executada com êxito O aplicativo que gera o relatório determina se algum dado deve ser incluído com o relatório.

Além de transmitir essa solicitação para o aplicativo recuperando a mensagem, o gerenciador de filas não executa nenhuma ação com base nessa opção. O aplicativo de recuperação deve gerar o relatório se apropriado.

### **MQRO\_NAN**

Esse tipo de relatório é gerado pelo aplicativo que recupera a mensagem e age sobre ela.. Isso indica que a ação solicitada na mensagem não foi executada com êxito O aplicativo que gera o relatório determina se algum dado deve ser incluído com o relatório. Por exemplo, talvez você queira incluir alguns dados indicando por que a solicitação não pôde ser executada.

Além de transmitir essa solicitação para o aplicativo recuperando a mensagem, o gerenciador de filas não executa nenhuma ação com base nessa opção. O aplicativo de recuperação deve gerar o relatório se apropriado.

O aplicativo deve determinar quais condições correspondem a uma ação positiva e quais correspondem a uma ação negativa. No entanto, se a solicitação tiver sido executada apenas parcialmente, gere um relatório NAN em vez de um relatório PAN, se solicitado. Cada condição possível deve corresponder a uma ação positiva ou a uma ação negativa, mas não a ambas.

**Opções do identificador de mensagens:** Especifique uma das opções listadas para controlar como o *MsgId* da mensagem de relatório (ou da mensagem de resposta) deve ser configurado.

### **MQRO\_NEW\_MSG\_ID**

Essa é a ação padrão e indica que, se um relatório ou resposta for gerado como resultado dessa mensagem, um novo *MsgId* será gerado para a mensagem de relatório ou de resposta

### **MQRO\_PASS\_MSG\_ID**

Se um relatório ou resposta for gerado como resultado dessa mensagem, o *MsgId* dessa mensagem será copiado para o *MsgId* da mensagem de relatório ou resposta.

O *MsgId* de uma mensagem de publicação será diferente para cada assinante que receber uma cópia da publicação e, portanto, o *MsgId* copiado na mensagem de relatório ou de resposta será diferente para cada um.

Se essa opção não for especificada, MQRO\_NEW\_MSG\_ID será assumido

**Opções do identificador de correlação:** especifique uma das opções listadas para controlar como o *CorrelId* da mensagem de relatório (ou da mensagem de resposta) deve ser configurado.

#### **MQRO\_COPY\_MSG\_ID\_TO\_CORREL\_ID**

Esta é a ação padrão e indica que se um relatório ou resposta for gerado como resultado dessa mensagem, o *MsgId* dessa mensagem será copiado para o *CorrelId* da mensagem de relatório ou resposta.

O *MsgId* de uma mensagem de publicação será diferente para cada assinante que recebe uma cópia da publicação e, portanto, o *MsgId* copiado no *CorrelId* da mensagem de relatório ou de resposta será diferente para cada um.

#### **MQRO\_PASS\_CORREL\_ID**

Se um relatório ou resposta for gerado como resultado dessa mensagem, o *CorrelId* dessa mensagem será copiado para o *CorrelId* da mensagem de relatório ou resposta.

O *CorrelId* de uma mensagem de publicação será específico para um assinante, a menos que ele use a opção MQSO\_SET\_CORREL\_ID e configure o campo de ID SubCorrelno MQSD para MQCI\_NONE. Portanto, é possível que o *CorrelId* copiado no *CorrelId* da mensagem de relatório ou de resposta seja diferente para cada um.

Se essa opção não for especificada, MQRO\_COPY\_MSG\_ID\_TO\_CORREL\_ID será assumido.

Os servidores respondendo a solicitações ou gerando mensagens de relatório devem verificar se as opções MQRO\_PASS\_MSG\_ID ou MQRO\_PASS\_CORREL\_ID foram configuradas na mensagem original. Se eram, os servidores devem executar a ação descrita para essas opções. Se nenhum deles estiver configurado, os servidores deverão executar a ação padrão correspondente

**Opções de disposição:** especifique uma das opções listadas para controlar a disposição da mensagem original quando ela não puder ser entregue na fila de destino. O aplicativo pode configurar as opções de disposição independentemente de solicitar relatórios de exceção.

#### **MQRO\_DEAD\_LETTER\_Q**

Esta é a ação padrão e coloca a mensagem na fila de mensagens não entregues se a mensagem não puder ser entregue para a fila de destino Isso acontece nas seguintes situações:

- Quando o aplicativo que colocou a mensagem original não pode ser notificado de forma síncrona sobre o problema por meio do código de razão retornado pela chamada MQPUT ou MQPUT1 . Uma mensagem de relatório de exceção é gerada, se uma foi solicitada pelo emissor.
- Quando o aplicativo que colocou a mensagem original estava colocando em um tópico

#### **MQRO\_DISCARD\_MSG**

Isso descarta a mensagem se ela não puder ser entregue para a fila de destino Isso acontece nas seguintes situações:

- Quando o aplicativo que colocou a mensagem original não pode ser notificado de forma síncrona sobre o problema por meio do código de razão retornado pela chamada MQPUT ou MQPUT1 . Uma mensagem de relatório de exceção é gerada, se uma foi solicitada pelo emissor.
- Quando o aplicativo que colocou a mensagem original estava colocando em um tópico

Se você desejar retornar a mensagem original para o emissor, sem que a mensagem original seja colocada na fila de mensagens não entregues, o emissor deverá especificar MQRO\_DISCARD\_MSG com MQRO\_EXCEPTION\_WITH\_FULL\_DATA

#### **MQRO\_PASS\_DISCARD\_AND\_EXPIRY**

Se essa opção for configurada em uma mensagem e um relatório ou resposta for gerado devido a ela, o descritor de mensagens do relatório herdará:

- MQRO\_DISCARD\_MSG se foi configurado.

- O tempo de expiração restante da mensagem (se este não for um relatório de expiração) Se este for um relatório de expiração, o tempo de expiração será configurado como 60 segundos

### Opção de atividade

#### **MQRO\_ACTIVITY**

O uso desse valor permite que a rota de **qualquer** mensagem seja rastreada em uma rede do gerenciador de filas. A opção de relatório pode ser especificada em qualquer mensagem do usuário atual, permitindo que você comece a calcular a rota da mensagem através da rede.

Se o aplicativo que está gerando a mensagem não puder ativar a geração de relatórios de atividades, o relatório poderá ser ativado usando uma saída cruzada da API fornecida pelos administradores do gerenciador de filas.

#### **Nota:**

1. Quanto menos os gerenciadores de filas na rede forem capazes de gerar relatórios de atividades, menos detalhada será a rota.
2. Os relatórios de atividades podem ser difíceis de colocar na ordem correta para determinar a rota tomada.
3. Os relatórios de atividades podem não conseguir localizar uma rota para seu destino solicitado.
4. As mensagens com esse conjunto de opções de relatório devem ser aceitas por qualquer gerenciador de fila, mesmo se elas não entenderem a opção Isso permite que a opção de relatório seja configurada em qualquer mensagem do usuário, mesmo se ela for processada por um gerenciador de filas não-IBM WebSphere MQ 6.0 ou posterior
5. Se um processo, um gerenciador de filas ou um processo do usuário, executar uma atividade em uma mensagem com essa opção configurada, ele poderá escolher gerar e colocar um relatório de atividades.

**Opção padrão:** especifique o seguinte se nenhuma opção de relatório for necessária::

#### **MQRO\_NONE**

Utilize este valor para indicar que nenhuma outra opção foi especificada.. MQRO\_NONE é definido para a documentação do programa de auxílio Não se pretende que essa opção seja usada com qualquer outra, mas como seu valor é zero, tal uso não pode ser detectado.

#### **Informações gerais:**

1. Todos os tipos de relatório necessários devem ser solicitados especificamente pelo aplicativo enviando a mensagem original. Por exemplo, se um relatório COA for solicitado, mas um relatório de exceção não for, um relatório COA será gerado quando a mensagem for colocada na fila de destino, mas nenhum relatório de exceção será gerado se a fila de destino estiver cheia quando a mensagem chegar lá. Se nenhuma opção *Report* for configurada, nenhuma mensagem de relatório será gerada pelo gerenciador de filas ou agente do canal de mensagens (MCA).

Algumas opções de relatório podem ser especificadas mesmo que o gerenciador de fila local não as reconheça; isso é útil quando a opção deve ser processada pelo gerenciador de fila de *destino* Consulte [“Opções de relatório e sinalizações de mensagem”](#) na página 935 para obter mais detalhes.

Se uma mensagem de relatório for solicitada, o nome da fila para a qual o relatório será enviado deverá ser especificado no campo *ReplyToQ* Quando uma mensagem de relatório é recebida, a natureza do relatório pode ser determinada examinando o campo *Feedback* no descritor de mensagens

2. Se o gerenciador de filas ou MCA que gera uma mensagem de relatório não puder colocar a mensagem de relatório na fila de resposta (por exemplo, porque a fila de respostas ou a fila de transmissão está cheia), a mensagem de relatório será colocada na fila de mensagens não entregues. Se esse *também* falhar ou não houver fila de mensagens não entregues, a ação tomada dependerá do tipo da mensagem de relatório:
  - Se a mensagem de relatório for um relatório de exceção, a mensagem que gerou o relatório de exceção será deixada em sua fila de transmissão; isso assegura que a mensagem não seja perdida.

- Para todos os outros tipos de relatório, a mensagem de relatório é descartada e o processamento continua normalmente. Isso é feito porque a mensagem original já foi entregue com segurança (para mensagens de relatório COA ou COD) ou não é mais de interesse (para uma mensagem de relatório de expiração).

Quando uma mensagem de relatório for colocada com êxito em uma fila (a fila de destino ou uma fila de transmissão intermediária), a mensagem não estará mais sujeita a processamento especial; ela será tratada como qualquer outra mensagem.

3. Quando o relatório é gerado, a fila *ReplyToQ* é aberta e a mensagem de relatório é colocada usando a autoridade do *UserIdentifier* no MQMD da mensagem que está causando o relatório, exceto nos seguintes casos:

- Os relatórios de exceção gerados por um MCA de recebimento são colocados com qualquer autoridade que o MCA usou quando tentou colocar a mensagem que está causando o relatório.
- Os relatórios COA gerados pelo gerenciador de fila são colocados com qualquer autoridade que foi usada quando a mensagem que causou o relatório foi colocada no gerenciador de filas que gerou o relatório. Por exemplo, se a mensagem foi colocada por um MCA de recebimento usando o identificador de usuário do MCA, o gerenciador de filas coloca o relatório COA usando o identificador de usuário do MCA.

Os aplicativos que geram relatórios devem usar a mesma autoridade que eles usam para gerar uma resposta; geralmente, essa é a autoridade do identificador de usuário na mensagem original.

Se o relatório tiver que viajar para um destino remoto, os remetentes e receptores podem decidir se o aceitam, da mesma forma que o fazem para outras mensagens.

4. Se uma mensagem de relatório com dados for solicitada:

- A mensagem de relatório é sempre gerada com a quantidade de dados solicitada pelo emissor da mensagem original. Se a mensagem de relatório for muito grande para a fila de resposta, o processamento descrito acima ocorrerá; a mensagem de relatório nunca será truncada para caber na fila de respostas.
- Se o *Format* da mensagem original for MQFMT\_XMIT\_Q\_HEADER, os dados incluídos no relatório não incluem MQXQH. Os dados do relatório iniciam com o primeiro byte dos dados além do MQXQH na mensagem original. Isso ocorre independentemente de a fila ser ou não uma fila de transmissão.

5. Se uma mensagem de relatório de COA, COD ou expiração for recebida na fila de resposta, será garantido que a mensagem original chegou, foi entregue ou expirou, conforme apropriado. No entanto, se uma ou mais dessas mensagens de relatório forem solicitadas e não forem recebidas, o reverso não poderá ser assumido, pois pode ter ocorrido uma das seguintes situações:

- a. A mensagem de relatório é retida porque um link está inativo.
- b. A mensagem de relatório é retida porque existe uma condição de bloqueio em uma fila de transmissão intermediária ou na fila de resposta (por exemplo, a fila está cheia ou inibida para puts).
- c. A mensagem de relatório está em uma fila de mensagens não entregues.
- d. Quando o gerenciador de filas estava tentando gerar a mensagem de relatório, ele não pôde colocá-la na fila apropriada, nem na fila de mensagens não entregues, portanto, a mensagem de relatório não pôde ser gerada.
- e. Ocorreu uma falha do gerenciador de filas entre a ação sendo relatada (chegada, entrega ou expiração) e a geração da mensagem de relatório correspondente. (Isso não acontece para mensagens de relatório COD se o aplicativo recuperar a mensagem original em uma unidade de trabalho, pois a mensagem de relatório COD é gerada na mesma unidade de trabalho.)

As mensagens de relatório de exceções podem ser mantidas da mesma maneira pelos motivos 1, 2 e 3 acima. No entanto, quando um MCA não pode gerar uma mensagem de relatório de exceção (a mensagem de relatório não pode ser colocada na fila de resposta ou na fila de devoluções), a mensagem original permanece na fila de transmissão no emissor e o canal é fechado. Isso ocorre independentemente de a mensagem de relatório ser gerada na extremidade de envio ou de recebimento do canal.

6. Se a mensagem original for temporariamente bloqueada (resultando em uma mensagem de relatório de exceções sendo gerada e a mensagem original sendo colocada em uma fila de mensagens não entregues), mas o bloqueio for limpo e um aplicativo ler a mensagem original da fila de mensagens não entregues e colocá-la novamente em seu destino, o seguinte poderá ocorrer:
  - Mesmo que uma mensagem de relatório de exceção tenha sido gerada, a mensagem original eventualmente chega com êxito a seu destino
  - Mais de uma mensagem de relatório de exceção é gerada em relação a uma única mensagem original, porque a mensagem original pode encontrar outro bloqueio posteriormente..

#### **Mensagens de relatório ao colocar em um tópico:**

1. Os relatórios podem ser gerados ao colocar uma mensagem em um tópico Esta mensagem será enviada para todos os assinantes do tópico, que pode ser zero, um ou muitos. Isso deve ser levado em consideração ao optar por usar opções de relatório, pois muitas mensagens de relatório poderiam ser geradas como resultado...
2. Ao colocar uma mensagem em um tópico, pode haver muitas filas de destino que devem receber uma cópia da mensagem. Se algumas dessas filas de destino tiverem um problema, como fila cheia, a conclusão bem-sucedida do MQPUT dependerá da configuração de NPMGDLV ou PMSGDLV (dependendo da persistência da mensagem). Se a configuração for tal que a entrega de mensagens para a fila de destino deve ser bem-sucedida (por exemplo, é uma mensagem persistente para um assinante durável e PMSGDLV é configurado como ALL ou ALLDUR), o sucesso será definido como um dos seguintes critérios que estão sendo atendidos:
  - Colocação bem-sucedida na fila de assinantes
  - Uso de MQRO\_DEAD\_LETTER\_Q e uma colocação bem-sucedida na fila de Devoluções se a fila de assinantes não puder receber a mensagem
  - Use MQRO\_DISCARD\_MSG se a fila de assinantes não puder receber a mensagem.

#### **Mensagens de relatório para segmentos de mensagens:**

1. Mensagens de relatório podem ser solicitadas para mensagens que têm segmentação permitida (consulte a descrição do sinalizador MQMF\_SEGMENTATION\_ALLOWED). Se o gerenciador de filas achar necessário segmentar a mensagem, uma mensagem de relatório poderá ser gerada para cada um dos segmentos que subsequentemente encontrar a condição relevante. Os aplicativos devem estar preparados para receber várias mensagens de relatório para cada tipo de mensagem de relatório solicitada Use o campo *GroupId* na mensagem de relatório para correlacionar os diversos relatórios com o identificador de grupo da mensagem original e o campo *Feedback* identificar o tipo de cada mensagem de relatório.
2. Se MQGMO\_LOGICAL\_ORDER for usado para recuperar mensagens de relatório para segmentos, esteja ciente de que os relatórios de *diferentes tipos* podem ser retornados pelas chamadas MQGET sucessivas Por exemplo, se os relatórios COA e COD forem solicitados para uma mensagem que é segmentada pelo gerenciador de filas, as chamadas MQGET para as mensagens de relatório poderão retornar as mensagens de relatório COA e COD intercaladas de uma maneira imprevisível. Evite isso usando a opção MQGMO\_COMPLETE\_MSG (opcionalmente com MQGMO\_ACCEPT\_TRUNCATED\_MSG). MQGMO\_COMPLETE\_MSG faz o gerenciador de filas remontar mensagens de relatório que possuem o mesmo tipo de relatório. Por exemplo, a primeira chamada MQGET pode remontar todas as mensagens COA relacionadas à mensagem original e a segunda chamada MQGET pode remontar todas as mensagens COD. Qual é remontado primeiro depende de qual tipo de mensagem de relatório ocorre primeiro na fila.
3. Os próprios aplicativos que colocam segmentos podem especificar diferentes opções de relatório para cada segmento. No entanto, observe os seguintes pontos:
  - Se os segmentos forem recuperados usando a opção MQGMO\_COMPLETE\_MSG, apenas as opções de relatório no *primeiro* segmento serão consideradas pelo gerenciador de filas.
  - Se os segmentos forem recuperados um por um e a maioria deles tiver uma das opções MQRO\_COD\_\*, mas pelo menos um segmento não, não será possível usar a opção MQGMO\_COMPLETE\_MSG para recuperar as mensagens de relatório com uma única chamada MQGET ou usar a opção

MQGMO\_ALL\_SEGMENTS\_AVAILABLE para detectar quando todas as mensagens de relatório chegaram.

- Em uma rede do MQ, os gerenciadores de filas podem ter recursos diferentes. Se uma mensagem de relatório para um segmento for gerada por um gerenciador de filas ou MCA que não suporta segmentação, o gerenciador de fila ou MCA não incluirá, por padrão, as informações de segmento necessárias na mensagem de relatório e isso poderá dificultar a identificação da mensagem original que causou a geração do relatório. Evite essa dificuldade solicitando dados com a mensagem de relatório, ou seja, especificando as opções MQRO\_\*\_WITH\_DATA ou MQRO\_\*\_WITH\_FULL\_DATA apropriadas. No entanto, esteja ciente de que, se MQRO\_\*\_WITH\_DATA for especificado, *menos de 100 bytes* de dados da mensagem do aplicativo poderão ser retornados para o aplicativo que recupera a mensagem de relatório, se a mensagem de relatório for gerada por um gerenciador de fila ou MCA que não suporta segmentação.

**Conteúdo do descritor de mensagens para uma mensagem de relatório:** Quando o gerenciador de fila ou o agente do canal de mensagens (MCA) gera uma mensagem de relatório, ele configura os campos no descritor de mensagem para os valores a seguir e, em seguida, coloca a mensagem de maneira normal..

Tabela 501. Valores utilizados para campos MQMD quando uma mensagem de relatório é gerada pelo sistema

<b>Campo no MQMD</b>	<b>Valor Usado</b>
<i>StrucId</i>	MQMD_STRUC_ID
<i>Version</i>	MQMD_VERSION_2
<i>Report</i>	MQRO_NONE
<i>MsgType</i>	MQMT_REPORT
<i>Expiry</i>	MQEI_UNLIMITED
<i>Feedback</i>	Conforme apropriado para a natureza do relatório (MQFB_COA, MQFB_COD, MQFB_EXPIRATION ou um valor MQRC_*)
<i>Encoding</i>	Copiado do descritor de mensagens original
<i>CodedCharSetId</i>	Copiado do descritor de mensagens original
<i>Format</i>	Copiado do descritor de mensagens original
<i>Priority</i>	Copiado do descritor de mensagens original
<i>Persistence</i>	Copiado do descritor de mensagens original
<i>MsgId</i>	Conforme especificado pelas opções de relatório no descritor de mensagem original..
<i>CorrelId</i>	Conforme especificado pelas opções de relatório no descritor de mensagem original..
<i>BackoutCount</i>	0
<i>ReplyToQ</i>	Espaços em branco
<i>ReplyToQMGr</i>	Nome do gerenciador de filas
<i>UserIdentifier</i>	conforme configurado pela opção MQPMO_PASS_IDENTITY_CONTEXT
<i>AccountingToken</i>	conforme configurado pela opção MQPMO_PASS_IDENTITY_CONTEXT
<i>ApplIdentityData</i>	conforme configurado pela opção MQPMO_PASS_IDENTITY_CONTEXT
<i>PutApplType</i>	MQAT_QMGR ou conforme apropriado para o agente do canal de mensagens
<i>PutApplName</i>	Primeiros 28 bytes do nome do gerenciador de filas ou nome do agente do canal de mensagens. Para mensagens de relatório geradas pela ponte IMS, esse campo contém o nome do grupo XCF e o nome do membro XCF do sistema IMS ao qual a mensagem se relaciona

Tabela 501. Valores utilizados para campos MQMD quando uma mensagem de relatório é gerada pelo sistema (continuação)

<b>Campo no MQMD</b>	<b>Valor Usado</b>
<i>PutDate</i>	Data quando a mensagem de relatório é enviada
<i>PutTime</i>	Hora em que a mensagem de relatório é enviada..
<i>ApplOriginData</i>	Espaços em branco
<i>GroupId</i>	Copiado do descritor de mensagens original
<i>MsgSeqNumber</i>	Copiado do descritor de mensagens original
<i>Offset</i>	Copiado do descritor de mensagens original
<i>MsgFlags</i>	Copiado do descritor de mensagens original
<i>OriginalLength</i>	Copiado do descritor de mensagens original, se não for MQOL_UNDEFINED, e configurado para o comprimento dos dados da mensagem original, caso contrário,

Um aplicativo que gera um relatório é recomendado para configurar valores semelhantes, exceto para o seguinte:

- O campo *ReplyToQMGr* pode ser configurado para espaços em branco (o gerenciador de filas muda isso para o nome do gerenciador de fila local quando a mensagem é colocada)..
- Configure os campos de contexto usando a opção que teria sido usada para uma resposta, normalmente MQPMO\_PASS\_IDENTITY\_CONTEXT.

**Analisando o campo de relatório:** O campo *Report* contém subcampos; por isso, os aplicativos que precisam verificar se o emissor da mensagem solicitou um relatório específico devem usar uma das técnicas descritas em [“Analisando o campo de relatório”](#) na página 936.

Esse é um campo de saída para a chamada MQGET e um campo de entrada para as chamadas MQPUT e MQPUT1 . O valor inicial deste campo é MQRO\_NONE.

### **MsgType (MQLONG) para MQMD**

Isso indica o tipo da mensagem. Os tipos de mensagens são agrupados da seguinte forma:

#### **MQMT\_SYSTEM\_FIRST**

Valor mais baixo para tipos de mensagens definidas pelo sistema.

#### **MQMT\_SYSTEM\_LAST**

Valor mais alto para tipos de mensagens definidos pelo sistema.

Os valores a seguir estão definidos atualmente no intervalo do sistema:

#### **MQMT\_DATAGRAM**

A mensagem é uma que não requer resposta.

#### **MQMT\_REQUEST**

A mensagem é aquela que requer uma resposta.

Especifique o nome da fila à qual enviar a resposta no campo *ReplyToQ* . O campo *Report* indica como configurar *MsgId* e *CorrelId* da resposta.

#### **MQMT\_REPLY**

A mensagem é a resposta a uma mensagem de solicitação anterior (MQMT\_REQUEST) A mensagem deve ser enviada para a fila indicada pelo campo *ReplyToQ* da mensagem de solicitação Use o campo *Report* da solicitação para controlar como configurar *MsgId* e *CorrelId* da resposta.

**Nota:** O gerenciador de filas não impinge o relacionamento de solicitação-resposta; esta é uma responsabilidade do aplicativo

## **MQMT\_REPORT**

A mensagem está relatando alguma ocorrência esperada ou inesperada, geralmente relacionada a alguma outra mensagem (por exemplo, foi recebida uma mensagem de solicitação que continha dados inválidos). Envie a mensagem para a fila indicada pelo campo *ReplyToQ* do descritor de mensagem da mensagem original Configure o campo *Feedback* s para indicar a natureza do relatório Use o campo *Report* da mensagem original para controlar como configurar *MsgId* e *CorrelId* da mensagem de relatório.

As mensagens de relatório geradas pelo gerenciador de filas ou agente do canal de mensagem são sempre enviadas para a fila *ReplyToQ*, com os campos *Feedback* e *CorrelId* configurados conforme descrito acima.

Os valores definidos pelo aplicativo também podem ser usados Eles devem estar dentro do seguinte intervalo:

## **MQMT\_APPL\_FIRST**

Valor mais baixo para tipos de mensagens definidas pelo aplicativo.

## **MQMT\_APPL\_LAST**

Valor mais alto para tipos de mensagens definidas pelo aplicativo.

Para as chamadas MQPUT e MQPUT1, o valor *MsgType* deve estar dentro do intervalo definido pelo sistema ou do intervalo definido pelo aplicativo; se não estiver, a chamada falhará com o código de razão MQRC\_MSG\_TYPE\_ERROR

Esse é um campo de saída para a chamada MQGET e um campo de entrada para as chamadas MQPUT e MQPUT1. O valor inicial desse campo é MQMT\_DATAGRAM.

## **Expiração (MQLONG) para MQMD**

Este é um período de tempo expresso em décimos de segundo, definido pelo aplicativo que coloca a mensagem. A mensagem se torna elegível para ser descartada, se não tiver sido removida da fila de destino antes de decorrer este período de tempo.

Por exemplo, para configurar um minuto para o tempo de expiração, é necessário configurar **MQMDEpiry** a 600.

O valor é decrementado para refletir o tempo que a mensagem gasta na fila de destino e também em quaisquer filas de transmissão intermediárias se a entrada for para uma fila remota... Ele também pode ser reduzido por agentes do canal de mensagens para refletir os tempos de transmissão, se eles forem significativos. Da mesma forma, um aplicativo que encaminhar essa mensagem para outra fila poderá diminuir o valor, se necessário, se ele tiver retido a mensagem por um tempo significativo.. No entanto, o prazo de expiração é tratado como aproximado e o valor não precisa ser reduzido para refletir pequenos intervalos de tempo.

Quando a mensagem é recuperada por um aplicativo usando a chamada MQGET, o campo *Expiry* representa o tempo de expiração que ainda permanece.

Após o tempo de expiração de uma mensagem, ela se torna elegível para ser descartada pelo gerenciador de filas. A mensagem é descartada quando ocorre uma chamada MQGET de procura ou não de procura que teria retornado a mensagem se ela ainda não tivesse expirado Por exemplo, uma chamada MQGET não de navegação com o campo *MatchOptions* em MQGMO configurado como MQMO\_NONE leitura de uma fila ordenada FIFO descarta todas as mensagens expiradas até a primeira mensagem não expirada. Com uma fila ordenada de prioridade, a mesma chamada descartará mensagens expiradas de prioridade mais alta e mensagens de prioridade igual que chegaram na fila antes da primeira mensagem não expirada.

Uma mensagem que expirou nunca é retornada a um aplicativo (por uma chamada de procura ou uma chamada MQGET não de procura), portanto, o valor no campo *Expiry* do descritor de mensagens após uma chamada MQGET bem-sucedida é maior que zero ou o valor especial MQEI\_UNLIMITED.

Se uma mensagem for colocada em uma fila remota, a mensagem poderá expirar (e ser descartada) enquanto estiver em uma fila de transmissão intermediária, antes de a mensagem atingir a fila de destino.



Um relatório será gerado quando uma mensagem expirada for descartada, se a mensagem tiver especificado uma das opções de relatório MQRO\_EXPIRATION\_\*. Se nenhuma dessas opções for especificada, nenhum relatório será gerado; a mensagem será assumida como não mais relevante após esse período de tempo (talvez porque uma mensagem posterior a substituiu).

Para uma mensagem colocada no ponto de sincronização, o intervalo de expiração começa no momento em que a mensagem é colocada, não no momento em que o ponto de sincronização é confirmado. É possível que o intervalo de expiração possa passar antes que o ponto de sincronização seja confirmado. Neste caso, a mensagem será descartada em algum momento após a operação de consolidação, e a mensagem não será retornada a um aplicativo em resposta a uma operação MQGET

Qualquer outro programa que descarte mensagens com base no tempo de expiração também deve enviar uma mensagem de relatório apropriada se uma foi solicitada.

#### Notas:

1. Se uma mensagem for colocada com um tempo *Expiry* de zero ou um número maior que 999 999 999, a chamada MQPUT ou MQPUT1 falhará com o código de razão MQRC\_EXPIRY\_ERROR; nenhuma mensagem de relatório será gerada neste caso.

Para ativar o código de razão 2013, MQRC\_EXPIRY\_ERROR, você deve ativar a variável de ambiente AMQ\_ENFORCE\_MAX\_EXPIRY\_ERROR.

O seguinte usa um exemplo para Linux:

```
$ export AMQ_ENFORCE_MAX_EXPIRY_ERROR=True
```

Observe que:

- O importante é exportar a variável
  - O valor real é ignorado, no entanto, usar True pode ser útil ao revisar a configuração.
2. Como uma mensagem com um tempo de expiração decorrido pode não ser descartada até mais tarde, pode haver mensagens em uma fila que passaram seu tempo de expiração e, portanto, não são elegíveis para recuperação. No entanto, essas mensagens contam para o número de mensagens na fila para todos os propósitos, incluindo o acionamento de profundidade.  
  
Se um assinante / consumidor (cliente) tentar obter uma mensagem e essa mensagem tiver expirado, o cliente não receberá nada, pois a mensagem foi descartada porque era muito antiga. Além disso, o cliente não receberá nenhuma mensagem de erro
  3. Um relatório de expiração é gerado, se solicitado, quando a mensagem é descartada, não quando se torna elegível para descarte.
  4. Descartar uma mensagem expirada e gerar um relatório de expiração se solicitado, nunca faz parte da unidade de trabalho do aplicativo, mesmo se a mensagem foi planejada para descartar como resultado de uma chamada MQGET operando em uma unidade de trabalho.
  5. Se uma mensagem quase expirada for recuperada por uma chamada MQGET dentro de uma unidade de trabalho e a unidade de trabalho for restaurada subsequentemente, a mensagem poderá se tornar elegível para ser descartada antes que possa ser recuperada novamente
  6. Se uma mensagem quase expirada for bloqueada por uma chamada MQGET com MQGMO\_LOCK, a mensagem poderá se tornar elegível para ser descartada antes que possa ser recuperada por uma chamada MQGET com MQGMO\_MSG\_UNDER\_CURSOR; o código de razão MQRC\_NO\_MSG\_UNDER\_CURSOR será retornado nessa chamada MQGET subsequente se isso acontecer.
  7. Quando uma mensagem de solicitação com um tempo de expiração maior que zero é recuperada, o aplicativo pode executar uma das ações a seguir ao enviar a mensagem de resposta:
    - Copie o tempo de expiração restante da mensagem de pedido para a mensagem de resposta
    - Configure o tempo de expiração na mensagem de resposta para um valor explícito maior que zero.
    - Configure o tempo de expiração na mensagem de resposta para MQEI\_UNLIMITED

A ação a ser tomada depende do design do aplicativo. No entanto, a ação padrão para colocar mensagens em uma fila de mensagens não entregues (mensagens não entregues) deve ser preservar o tempo de expiração restante da mensagem e continuar a diminuí-la.

8. As mensagens do acionador são sempre geradas com MQEI\_UNLIMITED
9. Uma mensagem (normalmente em uma fila de transmissão) que possui um nome *Format* de MQFMT\_XMIT\_Q\_HEADER possui um segundo descritor de mensagens no MQXQH. Portanto, ele possui dois campos *Expiry* associados a ele.. Os seguintes pontos adicionais devem ser observados neste caso:
  - Quando um aplicativo coloca uma mensagem em uma fila remota, o gerenciador de filas coloca a mensagem inicialmente em uma fila de transmissão local e prefixa os dados da mensagem do aplicativo com uma estrutura MQXQH. O gerenciador de filas configura os valores dos dois campos *Expiry* para serem os mesmos especificados pelo aplicativo.  
  
Se um aplicativo colocar uma mensagem diretamente em uma fila de transmissão local, os dados da mensagem já deverão começar com uma estrutura MQXQH, e o nome do formato deverá ser MQFMT\_XMIT\_Q\_HEADER. Neste caso, o aplicativo não precisa configurar os valores desses dois campos *Expiry* para serem os mesmos (O gerenciador de filas verifica se o campo *Expiry* no MQXQH contém um valor válido e se os dados da mensagem são longos o suficiente para incluí-lo). Para um aplicativo que pode gravar diretamente na fila de transmissão, o aplicativo precisa criar um cabeçalho da fila de transmissão com o descritor de mensagens incorporado. No entanto, se o valor de expiração no descritor de mensagens gravado na fila de transmissão estiver inconsistente com o valor no descritor de mensagens integrado, ocorrerá uma rejeição de erro de expiração...
  - Quando uma mensagem com um nome *Format* de MQFMT\_XMIT\_Q\_HEADER é recuperada de uma fila (se essa for uma fila normal ou de transmissão), o gerenciador de filas decrementa *ambos* esses *Expiry* campos com o tempo gasto esperando na fila. Nenhum erro será emitido se os dados da mensagem não forem longos o suficiente para incluir o campo *Expiry* no MQXQH
  - O gerenciador de filas utiliza o campo *Expiry* no descritor de mensagem separado (ou seja, não aquele no descritor de mensagem integrado na estrutura MQXQH) para testar se a mensagem é elegível para descarte...
  - Se os valores iniciais dos dois campos *Expiry* forem diferentes, o tempo do *Expiry* no descritor de mensagens separado quando a mensagem é recuperada poderá ser maior que zero (portanto, a mensagem não é elegível para descarte), enquanto o tempo de acordo com o campo *Expiry* no MQXQH tiver decorrido.. Nesse caso, o campo *Expiry* no MQXQH é configurado como zero.
10. O tempo de expiração em uma mensagem de resposta retornada da ponte IMS é ilimitado, a menos que MQIIH\_PASS\_EXPIRATION seja configurado no campo Sinalizadores do MQIIH. Consulte [Sinalizadores](#) para obter mais informações

O valor especial a seguir é reconhecido:

#### **MQEI\_UNLIMITED**

A mensagem tem um prazo de expiração ilimitado

Esse é um campo de saída para a chamada MQGET e um campo de entrada para as chamadas MQPUT e MQPUT1. O valor inicial desse campo é MQEI\_UNLIMITED.

#### *Mensagens expiradas em z/OS*

No IBM MQ for z/OS, as mensagens que expiraram são descartados pela próxima chamada MQGET apropriada

No entanto, se nenhuma chamada ocorrer, a mensagem expirada não será descartada e, para algumas filas, um grande número de mensagens expiradas poderá ser acumulado. Para corrigir isso, configure o gerenciador de filas para varrer filas periodicamente e descartar mensagens expiradas em uma ou mais filas de uma das seguintes maneiras:

#### **Varredura Periódica**

É possível especificar um período usando o atributo do gerenciador de filas EXPRYINT (intervalo de expiração).. Cada vez que o intervalo de expiração é atingido, o gerenciador de fila procura filas candidatas que valem a pena varrer para descartar mensagens expiradas.

O gerenciador de filas mantém informações sobre as mensagens expiradas em cada fila e sabe se uma varredura para mensagens expiradas é válida. Portanto, apenas uma seleção de filas é varrida a qualquer momento

As filas compartilhadas são varridas por apenas um gerenciador de filas em um grupo de filas compartilhadas. Geralmente, é o primeiro gerenciador de filas a reiniciar ou o primeiro a ter EXPRYINT configurado. Se esse gerenciador de filas for finalizado, outro gerenciador de filas no grupo de filas compartilhadas assumirá a varredura da fila. Configure o valor do intervalo de expiração para todos os gerenciadores de filas em um grupo de filas compartilhadas para o mesmo valor.


Observe que o processamento de expiração ocorre para cada fila quando um gerenciador de filas é reiniciado, independentemente da configuração de EXPRYINT



### Solicitação Explícita

Emita o comando REFRESH QMGR TYPE (EXPIRE), especificando a fila ou as filas que você deseja varrer.

#### *Aplicando prazos de expiração mais baixos*

Os administradores podem limitar o tempo de expiração de qualquer mensagem colocada em uma fila ou tópico usando o atributo **CAEXPRY** especificado no atributo **CUSTOM** na fila ou tópico.



**Importante:**  Não será possível usar o atributo **CAEXPRY** introduzido em IBM MQ 9.3.1 com um cluster se o repositório completo estiver em z/OS

  Em IBM MQ 9.3.1, os administradores podem limitar o tempo de expiração usando o atributo **CAEXPRY** de uma fila ou tópico sem precisar especificá-lo dentro do atributo **CUSTOM**. Se uma fila ou um tópico já tiver o atributo **CAEXPRY** configurado no atributo **CUSTOM**, você deverá remover o atributo **CAEXPRY** do atributo **CUSTOM** antes que o novo atributo **CAEXPRY** seja modificado. É possível fazer isso em um único comando, por exemplo:

```
ALTER QLOCAL(Q1) CAEXPRY(1000) CUSTOM('')
```

**Nota:** Ao migrar um objeto de uma versão anterior do produto, seu valor **CAEXPRY** é configurado para o valor padrão de NOLIMIT. Se o atributo **CAEXPRY** foi configurado dentro do atributo **CUSTOM**, essa opção terá precedência

Se você desejar usar o novo atributo **CAEXPRY**, deverá remover o atributo **CAEXPRY** do atributo **CUSTOM** primeiro. Configurar ambas as opções não funciona.

  O atributo **CAEXPRY** revisado configurado diretamente em filas ou tópicos (não dentro do atributo **CUSTOM**) é um atributo em cluster. Observe que todas as instâncias de uma fila de clusters devem usar o mesmo valor para seu atributo **CAEXPRY**. Ainda é possível que uma fila de transmissão reduza o tempo de expiração de uma mensagem se **CAEXPRY** tiver sido configurado na fila de transmissão e o valor for inferior ao atributo **CAEXPRY** da fila de clusters.

Um tempo de expiração especificado no campo **Expiry** do MQMD, por um aplicativo, que é maior que o valor **CAEXPRY** especificado na fila ou no tópico será substituído por esse valor **CAEXPRY**. Um tempo de expiração especificado por um aplicativo, que é inferior ao valor **CAEXPRY**, será usado

Observe que o valor de **CAEXPRY** é expresso em décimos de segundos, portanto, um minuto possui um valor de 600.

Se mais de um objeto for usado no caminho de resolução, por exemplo quando uma mensagem for colocada em um alias ou fila remota, o menor de todos os valores **CAEXPRY** será usado como o limite superior para a expiração da mensagem.

Mudanças nos valores **CAEXPRY** entram em vigor imediatamente. O valor de expiração é avaliado para cada colocação em uma fila ou tópico e, portanto, é sensível à resolução do objeto, que pode ser diferente entre cada operação de colocação

No entanto, observe que as mensagens existentes na fila, antes de uma mudança em **CAEXPRY**, não são afetadas pela mudança (ou seja, seu tempo de expiração permanece intacto). Apenas as novas mensagens que são colocadas na fila após a mudança em **CAEXPRY** têm o novo tempo de expiração

Em um cluster no qual um put é executado em uma fila aberta com MQOO\_BIND\_NOT\_FIXED, as mensagens podem ser designadas a diferentes valores de expiração em cada put, dependendo do valor **CAPEXPY** configurado para a fila de transmissão usada pelo canal, que envia a mensagem para o gerenciador de filas de destino selecionado.

Observe que um put em uma fila ou tópico por um aplicativo JMS especificando um atraso de entrega falha com MQRC\_EXPIRY\_ERROR, se o atraso de entrega estiver além do tempo de expiração resolvido para a fila ou tópico de destino. Um atributo **CAPEXPY** configurado em uma fila resolvida para um destino JMS pode causar esse erro..

**Nota:** **CAPEXPY** não deve ser usado em nenhuma fila que conterá IBM MQ mensagens geradas internamente, como qualquer SYSTEM.CLUSTER.\* e SYSTEM.PROTECTION.POLICY.QUEUE.

### Referências relacionadas

[DEFINE para filas](#)

[DEFINE TOPIC](#)

### **Feedback (MQLONG) para MQMD**

O campo Feedback é utilizado com uma mensagem do tipo MQMT\_REPORT para indicar a natureza do relatório e é significativo apenas com esse tipo de mensagem.

O campo pode conter um dos valores MQFB\_\* ou um dos valores MQRC\_\*. Os códigos de feedback são agrupados da seguinte forma:

#### **MQFB\_NONE**

Nenhum feedback fornecido.

#### **MQFB\_SYSTEM\_FIRST**

Valor mais baixo para feedback gerado pelo sistema.

#### **MQFB\_SYSTEM\_LAST**

Valor mais alto para feedback gerado pelo sistema.

O intervalo de códigos de feedback gerados pelo sistema MQFB\_SYSTEM\_FIRST por meio do MQFB\_SYSTEM\_LAST inclui os códigos de feedback gerais listados neste tópico (MQFB\_\*) e também os códigos de razão (MQRC\_\*) que podem ocorrer quando a mensagem não pode ser colocada na fila de destino...

#### **MQFB\_APPL\_FIRST**

Valor mais baixo para feedback gerado pelo aplicativo.

#### **MQFB\_APPL\_LAST**

Valor mais alto para feedback gerado pelo aplicativo.

Os aplicativos que geram mensagens de relatório não devem usar códigos de feedback no intervalo do sistema (além de MQFB\_QUIT), a menos que desejem simular mensagens de relatório geradas pelo gerenciador de filas ou agente do canal de mensagem.

Nas chamadas MQPUT ou MQPUT1, o valor especificado deve ser MQFB\_NONE ou estar dentro do intervalo do sistema ou intervalo do aplicativo. Isso é verificado independentemente do valor de *MsgType*.

### **Códigos de feedback gerais:**

#### **MQFB\_COA**

Confirmação de chegada na fila de destino (consulte MQRO\_COA).

#### **MQFB\_COD**

Confirmação de entrega ao aplicativo de recebimento (consulte MQRO\_COD).

#### **MQFB\_EXPIRATION**

A mensagem foi descartada porque ela não tinha sido removida da fila de destino antes de seu tempo de expiração ter decorrido

#### **MQFB\_PAN**

Notificação de ação positiva (consulte MQRO\_PAN).

**MQFB\_NAN**

Notificação de ação negativa (consulte MQRO\_NAN).

**MQFB\_QUIT**

Terminar aplicativo.

Isso pode ser usado por um programa de planejamento de carga de trabalho para controlar o número de instâncias de um programa de aplicativo que estão em execução. Enviar uma mensagem MQMT\_REPORT com esse código de feedback para uma instância do programa de aplicativo indica para essa instância que ela deve parar o processamento. No entanto, a adesão a essa Convenção é uma questão para o aplicativo; ela não é imposta pelo gerenciador de filas.

**Códigos de feedback do canal:****MQFB\_CHANNEL\_COMPLETED**

Um canal terminou normalmente.

**MQFB\_CHANNEL\_FAIL**

Um canal terminou de forma anormal e entra no estado STOPPED.

**MQFB\_CHANNEL\_FAIL\_RETRY**

Um canal terminou de forma anormal e entra no estado RETRY.

**IMS-bridge códigos de feedback**

Esses códigos são usados quando um código de detecção inesperado do IMS-OTMA é recebido. O código de detecção ou, quando o código de detecção for 0x1A, o código de razão associado a esse código de detecção, é indicado no *Feedback*.

1. Para códigos *Feedback* no intervalo MQFB\_IMS\_FIRST (300) até MQFB\_IMS\_LAST (399), um código de detecção diferente de 0x1A foi recebido. O *sense code* é fornecido pela expressão (*Feedback* - MQFB\_IMS\_FIRST+1).
2. Para códigos *Feedback* no intervalo MQFB\_IMS\_NACK\_1A\_REASON\_FIRST (600) por meio de MQFB\_IMS\_NACK\_1A\_REASON\_LAST (855), um código de detecção de 0x1A foi recebido. O *código de razão* associado ao código de detecção é fornecido pela expressão (*Feedback* - MQFB\_IMS\_NACK\_1A\_REASON\_FIRST).

O significado dos códigos de detecção do IMS-OTMA e os códigos de razão correspondentes são descritos em *Open Transaction Manager Access Guide and Reference*.

Os códigos de feedback a seguir podem ser gerados pela ponte IMS :

**MQFB\_DATA\_LENGTH\_ZERO**

Um comprimento de segmento era zero nos dados do aplicativo da mensagem..

**MQFB\_DATA\_LENGTH\_NEGATIVE**

Um comprimento do segmento foi negativo nos dados do aplicativo da mensagem

**MQFB\_DATA\_LENGTH\_TOO\_BIG**

O comprimento de um segmento era muito grande nos dados do aplicativo da mensagem

**MQFB\_BUFFER\_OVERFLOW**

O valor de um dos campos de comprimento faria com que os dados sobrecarregassem o buffer de mensagens..

**MQFB\_LENGTH\_OFF\_BY\_ONE**

O valor de um dos campos de comprimento era 1 byte muito curto.

**MQFB\_IIH\_ERROR**

O campo *Format* no MQMD especifica MQFMT\_IMS, mas a mensagem não começa com uma estrutura MQIIH válida

**MQFB\_NOT\_AUTHORIZED\_FOR\_IMS**

O ID do usuário contido no descritor de mensagens MQMD ou a senha contida no campo *Authenticator* na estrutura MQIIH falhou na validação executada pela ponte IMS . Como resultado, a mensagem não foi transmitida para o IMS

**MQFB\_IMS\_ERROR**

Um erro inesperado foi retornado por IMS Consulte o log de erro do IBM MQ no sistema no qual a ponte IMS reside para obter mais informações sobre o erro

**MQFB\_IMS\_FIRST**

Quando o código de controle IMS-OTMA não é 0x1A, os códigos de feedback gerados pelo IMS estão no intervalo MQFB\_IMS\_FIRST (300) a MQFB\_IMS\_LAST (399). O próprio código de detecção do IMS-OTMA é *Feedback* menos MQFB\_IMS\_ERROR

**MQFB\_IMS\_LAST**

Valor mais alto para o feedback gerado pelo IMS quando o código de detecção não é 0x1A.

**MQFB\_IMS\_NACK\_1A\_REASON\_FIRST**

Quando o código de detecção é 0x1A, os códigos de feedback gerados pelo IMS estão no intervalo MQFB\_IMS\_NACK\_1A\_REASON\_FIRST (600) até MQFB\_IMS\_NACK\_1A\_REASON\_LAST (855).

**MQFB\_IMS\_NACK\_1A\_REASON\_LAST**

Valor mais alto para o feedback gerado pelo IMS quando o código de detecção é 0x1A

**CICS-bridge feedback codes:** os códigos de feedback a seguir podem ser gerados pelo CICS bridge:

**MQFB\_CICS\_APPL\_ABENDED**

O programa aplicativo especificado na mensagem terminou de forma anormal. Esse código de feedback ocorre apenas no campo *Reason* da estrutura MQDLH..

**MQFB\_CICS\_APPL\_NOT\_STARTED**

O EXEC CICS LINK para o programa de aplicativo especificado na mensagem falhou. Esse código de feedback ocorre apenas no campo *Reason* da estrutura MQDLH..

**MQFB\_CICS\_BRIDGE\_FAILURE**

CICS bridge finalizou de forma anormal sem concluir o processamento de erro normal.

**MQFB\_CICS\_CCSID\_ERROR**

O identificador do conjunto de caracteres não é válido.

**MQFB\_CICS\_CIH\_ERROR**

CICS estrutura do cabeçalho de informações ausente ou inválida.

**MQFB\_CICS\_COMMAREA\_ERROR**

O comprimento de CICS COMMAREA não é válido

**MQFB\_CICS\_CORREL\_ID\_ERROR**

Identificador de correlação inválido.

**MQFB\_CICS\_DLQ\_ERROR**

A tarefa CICS bridge não pôde copiar uma resposta para essa solicitação para a fila de mensagens não entregues. A solicitação foi restaurada..

**MQFB\_CICS\_ENCODING\_ERROR**

A codificação não é válida

**MQFB\_CICS\_INTERNAL\_ERROR**

CICS bridge encontrou um erro inesperado.

Esse código de feedback ocorre apenas no campo *Reason* da estrutura MQDLH..

**MQFB\_CICS\_NOT\_AUTHORIZED**

Identificador de usuário não autorizado ou senha inválida.

Esse código de feedback ocorre apenas no campo *Reason* da estrutura MQDLH..

**MQFB\_CICS\_UOW\_BACKED\_OUT**

A unidade de trabalho foi restaurada, por uma das seguintes razões:

- Uma falha foi detectada ao processar outra solicitação dentro da mesma unidade de trabalho.
- Ocorreu um encerramento anormal CICS enquanto a unidade de trabalho estava em andamento.

**MQFB\_CICS\_UOW\_ERROR**

O campo de controle da unidade de trabalho *UOWControl* não é válido

**Códigos de feedback da mensagem de rastreamento de rotas:****MQFB\_ACTIVITY**

Usado com o formato MQFMT\_EMBEDDED\_PCF para permitir a opção de dados do usuário após os relatórios de atividades

**MQFB\_MAX\_ACTIVIDADES**

Retornado quando a mensagem de rastreamento de rotas é descartada porque o número de atividades em que a mensagem foi envolvida excede o limite máximo de atividades

**MQFB\_NOT\_FORWARDED**

Retornado quando a mensagem de rastreamento de rotas é descartada porque ela está prestes a ser enviada para um gerenciador de filas remotas que não suporta mensagens de rastreamento de rotas

**MQFB\_NOT\_ENTREGUE**

Retornado quando a mensagem de rastreamento de rotas é descartada porque ela está prestes a ser colocada em uma fila local

**MQFB\_UNSUPPORTED\_FORWARDING**

Retornado quando a mensagem de rastreamento de rotas é descartada, porque um valor no parâmetro de encaminhamento não é reconhecido e está na máscara de bits rejeitada

**MQFB\_UNSUPPORTED\_DELIVERY**

Retornado quando a mensagem de rastreamento de rotas é descartada porque um valor no parâmetro de entrega não é reconhecido e está na máscara de bits rejeitada.

**IBM MQ códigos de razão:** Para mensagens de relatório de exceção, *Feedback* contém um código de razão IBM MQ. Entre os possíveis códigos de razão estão:

**MQRC\_PUT\_INHIBITED**

(2051, X'803 ') Chamadas de put inibidas para a fila

**MQRC\_Q\_FULL**

(2053, X'805 ') A fila já contém número máximo de mensagens.

**MQRC\_NOT\_AUTHORIZED**

(2035, X'7F3') Não autorizado para acesso.

**MQRC\_Q\_SPACE\_NOT\_AVAILABLE**

(2056, X'808 ') Nenhum espaço disponível no disco para fila.

**MQRC\_PERSISTENT\_NOT\_ALLOWED**

(2048, X'800 ') A Fila não suporta mensagens persistentes.

**MQRC\_MSG\_TOO\_BIG\_FOR\_Q\_MGR**

(2031, X'7EF') Comprimento da mensagem maior que o máximo para o gerenciador de filas.

**MQRC\_MSG\_TOO\_BIG\_FOR\_Q**

(2030, X'7EE') Comprimento da mensagem maior que o máximo para a fila.

Para obter uma lista completa de códigos de razão, consulte:

- Para IBM MQ for z/OS, consulte [conclusão da API e códigos de razão](#).
- Para todas as outras plataformas, consulte [conclusão da API e códigos de razão](#).

Esse é um campo de saída para a chamada MQGET e um campo de entrada para as chamadas MQPUT e MQPUT1. O valor inicial desse campo é MQFB\_NONE.

**Codificação (MQLONG) para MQMD**

Especifica a codificação numérica de dados numéricos na mensagem; ela não se aplica a dados numéricos na própria estrutura MQMD. A codificação numérica define a representação usada para números inteiros binários, números inteiros decimais compactados e números de vírgula flutuante..

No z/OS, a parte de número inteiro binário do campo `Encoding` também é usada para especificar a codificação de número inteiro de dados de caracteres no corpo da mensagem quando o identificador do conjunto de caracteres correspondente indica que a representação do conjunto de caracteres é dependente da codificação usada para números inteiros binários... Isso afeta apenas alguns conjuntos de caracteres multibyte (por exemplo, conjuntos de caracteres UTF-16)..

Na chamada `MQPUT` ou `MQPUT1`, o aplicativo deve configurar esse campo para o valor apropriado aos dados. O gerenciador de filas não verifica se o campo é válido.. O seguinte valor especial é definido:

#### **MQENC\_NATIVE**

A codificação é o padrão para a linguagem de programação e a máquina na qual o aplicativo está em execução

**Nota:** O valor desta constante depende da linguagem de programação e ambiente. Por esse motivo, os aplicativos devem ser compilados usando o cabeçalho, macro, arquivos `COPY` ou `INCLUDE` apropriados para o ambiente no qual o aplicativo será executado.

Aplicativos que colocam mensagens geralmente especificam `MQENC_NATIVE`. Os aplicativos que recuperam mensagens devem comparar esse campo com relação ao valor `MQENC_NATIVE`; se os valores forem diferentes, o aplicativo poderá precisar converter dados numéricos na mensagem Use a opção `MQGMO_CONVERT` para solicitar que o gerenciador de filas converta a mensagem como parte do processamento da chamada `MQGET` Consulte [“Codificações da máquina” na página 932](#) para obter detalhes de como o campo `Encoding` é construído..

Se você especificar a opção `MQGMO_CONVERT` na chamada `MQGET`, esse campo será um campo de entrada / saída.. O valor especificado pelo aplicativo é a codificação para a qual converter os dados da mensagem, se necessário. Se a conversão for bem-sucedida ou desnecessária, o valor será inalterado. Se a conversão for malsucedida, o valor após a chamada `MQGET` representa a codificação da mensagem não convertida que é retornada ao aplicativo.

Em outros casos, esse é um campo de saída para a chamada `MQGET` e um campo de entrada para as chamadas `MQPUT` e `MQPUT1`. O valor inicial desse campo é `MQENC_NATIVE`.

#### **CodedCharSetId (MQLONG) para MQMD**

Este campo especifica o identificador do conjunto de caracteres de dados de caracteres no corpo da mensagem.

**Nota:** Os dados de caractere em `MQMD` e as outras estruturas de dados do `MQ` que são parâmetros em chamadas devem estar no conjunto de caracteres do gerenciador de filas Isso é definido pelo atributo **CodedCharSetId** do gerenciador de filas; consulte [“Atributos do gerenciador de filas” na página 827](#) para obter detalhes desse atributo..

Se esse campo for configurado como `MQCCSI_Q_MGR` ao chamar `MQGET` com `MQGMO_CONVERT` nas opções, o comportamento será diferente entre aplicativos cliente e servidor. Para aplicativos do servidor, a página de códigos utilizada para conversão de caractere é o `CodedCharSetId` do gerenciador de filas; para aplicativos clientes, a página de códigos utilizada para conversão de caractere é a página de códigos do código do idioma atual

Para aplicativos clientes, `MQCCSI_Q_MGR` é preenchido, com base no código de idioma do cliente em vez daquele no gerenciador de filas. A exceção para essa regra é quando você coloca uma mensagem em uma fila de pontes `IMS`; o que é retornado, no campo `CodedCharSetId` de `MQMD`, é o `CCSID` do gerenciador de filas.

Você não deve usar o seguinte valor especial:

#### **MQCCSI\_APPL**

Isso resulta em um valor incorreto no campo `CodedCharSetId` do `MQMD` e causa um código de retorno de `MQRC_SOURCE_CCSID_ERROR` (ou `MQRC_FORMAT_ERROR` para `z/OS`) Quando a mensagem é recebida usando a chamada `MQGET` com a opção `MQGMO_CONVERT`

É possível usar os seguintes valores especiais:

#### **MQCCSI\_Q\_MGR**

Os dados de caracteres na mensagem estão no conjunto de caracteres do gerenciador de fila.



Nas chamadas MQPUT e MQPUT1 , o gerenciador de filas muda esse valor no MQMD que é enviado com a mensagem para o identificador do conjunto de caracteres verdadeiro do gerenciador de filas. Como resultado, o valor MQCCSI\_Q\_MGR nunca é retornado pela chamada MQGET..

### **MQCCSI\_DEFAULT**

O CodedCharSetId dos dados no campo *String* é definido pelo campo CodedCharSetId na estrutura do cabeçalho que precede a estrutura MQCFH, ou pelo campo CodedCharSetId no MQMD se o MQCFH estiver no início da mensagem

### **MQCCSI\_INHERIT**

Os dados de caractere na mensagem estão no mesmo conjunto de caracteres que esta estrutura; este é o conjunto de caracteres do gerenciador de filas (Somente para MQMD, MQCCSI\_INHERIT tem o mesmo significado que MQCCSI\_Q\_MGR.

O gerenciador de filas muda esse valor no MQMD que é enviado com a mensagem para o identificador do conjunto de caracteres real do MQMD... Desde que não ocorra nenhum erro, o valor MQCCSI\_INHERIT não será retornado pela chamada MQGET

Não use MQCCSI\_INHERIT se o valor do campo PutAppLType em MQMD for MQAT\_BROKER

### **MQCCSI\_EMBEDDED**

Os dados de caractere na mensagem estão em um conjunto de caracteres com o identificador que está contido nos próprios dados da mensagem... Pode haver qualquer número de identificadores de conjunto de caracteres integrados nos dados da mensagem, que se aplicam a diferentes partes dos dados.. Esse valor deve ser usado para mensagens PCF (com um formato de MQFMT\_ADMIN, MQFMT\_EVENT ou MQFMT\_PCF) que contêm dados em uma mistura de conjunto de caracteres... Cada estrutura MQCFST, MQCFSL e MQCFSF contida na mensagem PCF deve ter um identificador de conjunto de caracteres explícito especificado e não MQCCSI\_DEFAULT.

Se uma mensagem do formato MQFMT\_EMBEDDED\_PCF tiver que conter dados em uma mistura de conjuntos de caracteres, não use MQCCSI\_EMBEDDED Em vez disso, configure MQEPH\_CCSID\_EMBEDDED no campo Flags na estrutura MQEPH. Isso é equivalente à configuração de MQCCSI\_EMBEDDED na estrutura anterior Cada estrutura MQCFST, MQCFSL e MQCFSF contida na mensagem PCF deve ter um identificador de conjunto de caracteres explícito especificado e não MQCCSI\_DEFAULT. Para obter mais informações sobre a estrutura MQEPH, consulte [“MQEPH - cabeçalho PCF integrado”](#) na página 374..

Especifique este valor somente nas chamadas MQPUT e MQPUT1 . Se ele for especificado na chamada MQGET, ele evitará a conversão da mensagem

Nas chamadas MQPUT e MQPUT1 , o gerenciador de filas muda os valores MQCCSI\_Q\_MGR e MQCCSI\_INHERIT no MQMD que é enviado com a mensagem conforme descrito acima, mas não muda o MQMD especificado na chamada MQPUT ou MQPUT1 . Nenhuma outra verificação é executada no valor especificado

Os aplicativos que recuperam mensagens devem comparar esse campo com o valor que o aplicativo está esperando; se os valores forem diferentes, o aplicativo poderá precisar converter dados de caractere na mensagem.

Em z/OS, o campo Encoding do MQMD é usado para especificar a codificação de número inteiro de dados de caractere no corpo da mensagem, quando o campo CodedCharSetId do MQMD indica que a representação do conjunto de caracteres é dependente da codificação usada para números inteiros binários. Em Multiplataformas, a ordem de bytes de dados de caracteres é assumida como a mesma codificação de número inteiro nativo para a plataforma na qual o gerenciador de filas está em execução. Isso afeta apenas determinados conjuntos de caracteres multibyte (por exemplo, conjuntos de caracteres UTF-16 ).

Se você especificar a opção MQGMO\_CONVERT na chamada MQGET, esse campo será um campo de entrada / saída.. O valor especificado pelo aplicativo é o identificador do conjunto de caracteres codificados para o qual converter os dados da mensagem, se necessário. Se a conversão for bem-sucedida ou desnecessária, o valor será inalterado (exceto que o valor MQCCSI\_Q\_MGR ou MQCCSI\_INHERIT será convertido para o valor real) Se a conversão for malsucedida, o valor após a

chamada MQGET representa o identificador do conjunto de caracteres codificados da mensagem não convertida que é retornada ao aplicativo.

Caso contrário, este é um campo de saída para a chamada MQGET e um campo de entrada para as chamadas MQPUT e MQPUT1 . O valor inicial desse campo é MQCCSI\_Q\_MGR..

### **Formato (MQCHAR8) para MQMD**

Este é um nome que o remetente da mensagem utiliza para indicar ao destinatário a natureza dos dados na mensagem. Quaisquer caracteres que estejam no conjunto de caracteres do gerenciador de filas podem ser especificados para o nome, mas você deve restringir o nome para o seguinte:

- Maiúscula A a Z
- Dígitos numéricos de 0 a 9

Se outros caracteres forem usados, talvez não seja possível converter o nome entre os conjuntos de caracteres dos gerenciadores de filas de envio e recebimento.

Preencha o nome com espaços em branco até o comprimento do campo ou use um caractere nulo para finalizar o nome antes do final do campo; o nulo e quaisquer caracteres subsequentes são tratados como espaços em branco. Não especifique um nome com espaços em branco iniciais ou integrados. Para a chamada MQGET, o gerenciador de filas retorna o nome preenchido com espaços em branco para o comprimento do campo.

O gerenciador de filas não verifica se o nome está em conformidade com as recomendações descritas acima.

Os nomes que começam com MQ em maiúsculas, minúsculas e compostos por letras maiúsculas e minúsculas têm significados definidos pelo gerenciador de filas; não use nomes que começam com essas letras para seus próprios formatos. Os formatos integrados do gerenciador de filas são:

#### **MQFMT\_NONE**

A natureza dos dados é indefinida: os dados não podem ser convertidos quando a mensagem é recuperada de uma fila usando a opção MQGMO\_CONVERT.

Se você especificar MQGMO\_CONVERT na chamada MQGET e o conjunto de caracteres ou a codificação de dados na mensagem diferir daquele especificado no parâmetro **MsgDesc** , a mensagem será retornada com os seguintes códigos de conclusão e de razão (assumindo que nenhum outro erro):

- Código de conclusão MQCC\_WARNING e código de razão MQRC\_FORMAT\_ERROR se os dados MQFMT\_NONE estiverem no início da mensagem.
- Código de conclusão MQCC\_OK e código de razão MQRC\_NONE se os dados MQFMT\_NONE estiverem no final da mensagem (ou seja, precedido por uma ou mais estruturas de cabeçalho MQ). As estruturas do cabeçalho MQ são convertidas para o conjunto de caracteres solicitado e a codificação nesse caso.

Para a linguagem de programação C, a constante MQFMT\_NONE\_ARRAY também é definida; isso possui o mesmo valor que MQFMT\_NONE, mas é uma matriz de caracteres em vez de uma cadeia...


#### **MQFMT\_ADMIN**

A mensagem é uma mensagem de solicitação ou resposta do servidor de comando no formato de comando programável (PCF). As mensagens desse formato poderão ser convertidas se a opção MQGMO\_CONVERT for especificada na chamada MQGET. Consulte [Usando Formatos de Comando Programáveis](#) para obter mais informações sobre como usar mensagens de formato de comando programáveis.

Para a linguagem de programação C, a constante MQFMT\_ADMIN\_ARRAY também é definida; isso tem o mesmo valor que MQFMT\_ADMIN, mas é uma matriz de caracteres em vez de uma cadeia.

#### **MQFMT\_CICS**

Os dados da mensagem começam com o cabeçalho de informações CICS MQCIH, seguido pelos dados do aplicativo. O nome do formato dos dados do aplicativo é fornecido pelo campo `Format` na estrutura MQCIH.

 No z/OS, especifique a opção MQGMO\_CONVERT na chamada MQGET para converter mensagens que possuem o formato MQFMT\_CICS

Para a linguagem de programação C, a constante MQFMT\_CICS\_ARRAY também é definida; isso tem o mesmo valor que MQFMT\_CICS, mas é uma matriz de caracteres em vez de uma cadeia.

### **MQFMT\_COMMAND\_1**

A mensagem é uma mensagem de resposta do servidor de comandos MQSC que contém a contagem de objetos, o código de conclusão e o código de razão. As mensagens desse formato poderão ser convertidas se a opção MQGMO\_CONVERT for especificada na chamada MQGET.

Para a linguagem de programação C, a constante MQFMT\_COMMAND\_1\_ARRAY também é definida; isso tem o mesmo valor que MQFMT\_COMMAND\_1, mas é uma matriz de caracteres em vez de uma sequência.

### **MQFMT\_COMMAND\_2**

A mensagem é uma mensagem de resposta do servidor de comandos MQSC contendo informações sobre objetos solicitados. As mensagens desse formato poderão ser convertidas se a opção MQGMO\_CONVERT for especificada na chamada MQGET.

Para a linguagem de programação C, a constante MQFMT\_COMMAND\_2\_ARRAY também é definida; isso tem o mesmo valor que MQFMT\_COMMAND\_2, mas é uma matriz de caracteres em vez de uma sequência.

### **MQFMT\_DEAD\_LETTER\_HEADER**

Os dados da mensagem começam com o cabeçalho MQDLH. Os dados da mensagem original seguem imediatamente a estrutura MQDLH. O nome do formato dos dados da mensagem original é fornecido por o campo *Format* na estrutura MQDLH; consulte [“MQDLH - Cabeçalho de mensagens não entregues” na página 361](#) para obter detalhes dessa estrutura. As mensagens desse formato poderão ser convertidas se a opção MQGMO\_CONVERT for especificada na chamada MQGET.

Relatórios COA e COD não são gerados para mensagens que possuem um *Format* de MQFMT\_DEAD\_LETTER\_HEADER.

Para a linguagem de programação C, a constante MQFMT\_DEAD\_LETTER\_HEADER\_ARRAY também é definida; isso tem o mesmo valor que MQFMT\_DEAD\_LETTER\_HEADER, mas é uma matriz de caracteres em vez de uma sequência.

### **MQFMT\_DIST\_HEADER**

Os dados da mensagem começam com o cabeçalho MQDH da lista de distribuição; isso inclui as matrizes de registros MQOR e MQPMR. O cabeçalho da lista de distribuições pode ser seguido por dados adicionais. O formato dos dados adicionais (se houver) é fornecido pelo campo *Format* na estrutura MQDH; consulte [“MQDH - Cabeçalho de distribuição” na página 355](#) para obter detalhes dessa estrutura. Mensagens com formato MQFMT\_DIST\_HEADER podem ser convertidas se a opção MQGMO\_CONVERT for especificada na chamada MQGET.

Esse formato é suportado nos seguintes ambientes:

-  AIX
-  IBM i
-  Linux
-  Windows

e para o IBM MQ MQI clients conectado a esses sistemas

Para a linguagem de programação C, a constante MQFMT\_DIST\_HEADER\_ARRAY também é definida; ela tem o mesmo valor que MQFMT\_DIST\_HEADER, mas é uma matriz de caracteres em vez de uma sequência.

## **MQFMT\_EMBEDDED\_PCF**

Formato para uma mensagem de rastreamento de rotas, desde que o valor do comando PCF seja configurado como MQCMD\_TRACE\_ROUTE. O uso desse formato permite que os dados do usuário sejam enviados juntamente com a mensagem de rastreamento de rotas, desde que seus aplicativos possam lidar com os parâmetros PCF anteriores.

O cabeçalho PCF deve ser o primeiro cabeçalho, ou a mensagem não será tratada como uma mensagem de rastreamento de rotas. Isso significa que a mensagem não pode estar em um grupo e que mensagens de rastreamento de rotas não podem ser segmentadas. Se uma mensagem de rastreamento de rotas for enviada em um grupo, a mensagem será rejeitada com o código de razão MQRC\_MSG\_NOT\_ALLOWED\_IN\_GROUP.

Observe que MQFMT\_ADMIN também pode ser usado para o formato de uma mensagem de rastreamento de rotas, mas, neste caso, nenhum dado do usuário pode ser enviado junto com a mensagem de rastreamento de rotas.

## **MQFMT\_EVENT**

A mensagem é uma mensagem do evento MQ que relata um evento ocorrido. As mensagens de evento têm a mesma estrutura que os comandos programáveis; consulte [Mensagens de comando PCF](#) para obter mais informações sobre essa estrutura e [Monitoramento de eventos](#) para obter informações sobre eventos.

As mensagens do evento Version-1 poderão ser convertidas em todos os ambientes se a opção MQGMO\_CONVERT for especificada na chamada MQGET. As mensagens do evento Version-2 podem ser convertidas apenas em z/OS.

Para a linguagem de programação C, a constante MQFMT\_EVENT\_ARRAY também é definida; isso tem o mesmo valor que MQFMT\_EVENT, mas é uma matriz de caracteres em vez de uma cadeia.

## **MQFMT\_IMS**

Os dados da mensagem começam com o cabeçalho de informações MQIIH do IMS, que é seguido pelos dados do aplicativo. O nome do formato dos dados do aplicativo é fornecido pelo campo Format na estrutura MQIIH.

Para obter detalhes de como a estrutura MQIIH é manipulada ao usar MQGET com MQGMO\_CONVERT, consulte ["Formato \(MQCHAR8\) para MQIIH"](#) na página 423 e ["ReplyToFormat \(MQCHAR8\) para MQIIH"](#) na página 424.

Para a linguagem de programação C, a constante MQFMT\_IMS\_ARRAY também é definida; isso tem o mesmo valor que MQFMT\_IMS, mas é uma matriz de caracteres em vez de uma sequência.

## **MQFMT\_IMS\_VAR\_STRING**

A mensagem é uma sequência de variáveis IMS, que é uma sequência do formato 11zzccc, em que:

### **11**

é um campo de comprimento de 2 bytes especificando o comprimento total do item de sequência variável IMS. Esse comprimento é igual ao comprimento de 11 (2 bytes), mais o comprimento de zz (2 bytes), mais o comprimento da própria cadeia de caractere. 11 é um número inteiro binário de 2 bytes na codificação especificada pelo campo Encoding.

### **zz**

é um campo de 2 bytes contendo sinalizadores que são significativos para IMS. zz é uma sequência de bytes consistindo em dois campos MQBYTE e é transmitida sem mudança de emissor para receptor (ou seja, zz não está sujeito a nenhuma conversão).

### **ccc**

é uma sequência de caracteres de comprimento variável que contém 11-4 caracteres. ccc está no conjunto de caracteres especificado pelo campo CodedCharSetId ..

No z/OS, os dados da mensagem podem consistir em uma sequência de sequências de variáveis IMS unidas, com cada sequência no formato 11zzccc. Não deve haver bytes ignorados entre sequências variáveis IMS sucessivas. Isso significa que se a primeira sequência tiver um comprimento ímpar, a segunda sequência será desalinhada, ou seja, ela não começará em um limite que seja um múltiplo de

dois. Tome cuidado ao construir essas sequências em máquinas que requerem o alinhamento de tipos de dados elementares

Use a opção MQGMO\_CONVERT na chamada MQGET para converter mensagens que possuem o formato MQFMT\_IMS\_VAR\_STRING

Para a linguagem de programação C, a constante MQFMT\_IMS\_VAR\_STRING\_ARRAY também é definida; isso tem o mesmo valor que MQFMT\_IMS\_VAR\_STRING, mas é uma matriz de caracteres em vez de uma sequência.

### **MQFMT\_MD\_EXTENSION**

Os dados da mensagem começam com a extensão do descritor de mensagens MQMDE e são opcionalmente seguidos por outros dados (geralmente os dados da mensagem do aplicativo). O nome do formato, o conjunto de caracteres e a codificação dos dados que seguem o MQMDE são fornecidos pelos campos Format, CodedCharSetIde e Encoding no MQMDE. Consulte [“MQMDE-Extensão do descritor de mensagens”](#) na página 488 para obter detalhes sobre essa estrutura. As mensagens desse formato poderão ser convertidas se a opção MQGMO\_CONVERT for especificada na chamada MQGET.

Para a linguagem de programação C, a constante MQFMT\_MD\_EXTENSION\_ARRAY também é definida; isso tem o mesmo valor que MQFMT\_MD\_EXTENSION, mas é uma matriz de caracteres em vez de uma sequência.

### **MQFMT\_PCF**

A mensagem é uma mensagem definida pelo usuário em conformidade com a estrutura de uma mensagem de formato de comando programável (PCF). As mensagens desse formato poderão ser convertidas se a opção MQGMO\_CONVERT for especificada na chamada MQGET. Consulte [Usando Formatos de Comando Programáveis](#) para obter mais informações sobre como usar mensagens de formato de comando programáveis.

Para a linguagem de programação C, a constante MQFMT\_PCF\_ARRAY também é definida; ela tem o mesmo valor que MQFMT\_PCF, mas é uma matriz de caracteres em vez de uma sequência.

### **MQFMT\_REF\_MSG\_HEADER**

Os dados da mensagem começam com o cabeçalho da mensagem de referência MQRMH e são opcionalmente seguidos por outros dados. O nome do formato, o conjunto de caracteres e a codificação dos dados são fornecidos pelos campos Format, CodedCharSetIde e Encoding no MQRMH. Consulte [“MQRMH - cabeçalho de mensagem de referência”](#) na página 569 para obter detalhes sobre essa estrutura. As mensagens desse formato poderão ser convertidas se a opção MQGMO\_CONVERT for especificada na chamada MQGET.

Esse formato é suportado nos seguintes ambientes:

-  AIX
-  IBM i
-  Linux
-  Windows

e para o IBM MQ MQI clients conectado a esses sistemas

Para a linguagem de programação C, a constante MQFMT\_REF\_MSG\_HEADER\_ARRAY também é definida; isso tem o mesmo valor que MQFMT\_REF\_MSG\_HEADER, mas é uma matriz de caracteres em vez de uma cadeia.

### **MQFMT\_RF\_HEADER**

Os dados da mensagem começam com as regras e o cabeçalho de formatação MQRFH e são opcionalmente seguidos por outros dados. O nome do formato, o conjunto de caracteres e a codificação dos dados (se houver) são fornecidos pelos campos Format, CodedCharSetIde e Encoding no MQRFH. As mensagens desse formato poderão ser convertidas se a opção MQGMO\_CONVERT for especificada na chamada MQGET.

Para a linguagem de programação C, a constante MQFMT\_RF\_HEADER\_ARRAY também é definida; ela tem o mesmo valor que MQFMT\_RF\_HEADER, mas é uma matriz de caracteres em vez de uma sequência.

### **MQFMT\_RF\_HEADER\_2**

Os dados da mensagem começam com as regras version-2 e o cabeçalho de formatação MQRFH2e, opcionalmente, são seguidos por outros dados. O nome do formato, o conjunto de caractere e a codificação dos dados opcionais (se houver) são fornecidos pelos campos Format, CodedCharSetId e Encoding no MQRFH2. As mensagens desse formato poderão ser convertidas se a opção MQGMO\_CONVERT for especificada na chamada MQGET

Para a linguagem de programação C, a constante MQFMT\_RF\_HEADER\_2\_ARRAY também é definida; isso tem o mesmo valor que MQFMT\_RF\_HEADER\_2, mas é uma matriz de caracteres em vez de uma cadeia.

### **MQFMT\_STRING**

Os dados da mensagem do aplicativo podem ser uma sequência SBCS (conjunto de caracteres de byte único) ou uma sequência DBCS (conjunto de caracteres de byte duplo). As mensagens desse formato poderão ser convertidas se a opção MQGMO\_CONVERT for especificada na chamada MQGET

Para a linguagem de programação C, a constante MQFMT\_STRING\_ARRAY também é definida; isso tem o mesmo valor que MQFMT\_STRING, mas é uma matriz de caracteres em vez de uma cadeia.


### **MQFMT\_TRIGGER**

A mensagem é uma mensagem acionador, descrita pela estrutura MQTM; consulte [“MQTM-Mensagem do acionador”](#) na página 622 para obter detalhes dessa estrutura. As mensagens desse formato poderão ser convertidas se a opção MQGMO\_CONVERT for especificada na chamada MQGET

Para a linguagem de programação C, a constante MQFMT\_TRIGGER\_ARRAY também é definida; isso tem o mesmo valor que MQFMT\_TRIGGER, mas é uma matriz de caracteres em vez de uma sequência.

### **MQFMT\_WORK\_INFO\_HEADER**

Os dados da mensagem começam com o cabeçalho de informações de trabalho MQWIH, seguido pelos dados do aplicativo. O nome do formato dos dados do aplicativo é fornecido pelo campo Format na estrutura MQWIH.

 No z/OS, especifique a opção MQGMO\_CONVERT na chamada MQGET para converter os dados do usuário nas mensagens que possuem o formato MQFMT\_WORK\_INFO\_HEADER. No entanto, a própria estrutura MQWIH é sempre retornada no conjunto de caracteres e codificação do gerenciador de filas (ou seja, a estrutura MQWIH é convertida se a opção MQGMO\_CONVERT for especificada ou não).

Para a linguagem de programação C, a constante MQFMT\_WORK\_INFO\_HEADER\_ARRAY também é definida; isso tem o mesmo valor que MQFMT\_WORK\_INFO\_HEADER, mas é uma matriz de caracteres em vez de uma sequência.

### **MQFMT\_XMIT\_Q\_HEADER**

Os dados da mensagem começam com o cabeçalho da fila de transmissão MQXQH.. Os dados da mensagem original seguem imediatamente a estrutura MQXQH.. O nome do formato dos dados da mensagem original é fornecido pelo campo Format na estrutura MQMD, que faz parte do cabeçalho da fila de transmissão MQXQH.. Consulte [“MQXQH-Cabeçalho da fila de transmissão”](#) na página 641 para obter detalhes sobre essa estrutura

Relatórios COA e COD não são gerados para mensagens que possuem um Format de MQFMT\_XMIT\_Q\_HEADER.

Para a linguagem de programação C, a constante MQFMT\_XMIT\_Q\_HEADER\_ARRAY também é definida; isso tem o mesmo valor que MQFMT\_XMIT\_Q\_HEADER, mas é uma matriz de caracteres em vez de uma sequência.

Esse é um campo de saída para a chamada MQGET e um campo de entrada para as chamadas MQPUT e MQPUT1. O comprimento desse campo é fornecido por MQ\_FORMAT\_LENGTH. O valor inicial desse campo é MQFMT\_NONE.



## **Prioridade (MQLONG) para MQMD**

Para as chamadas MQPUT e MQPUT1 , o valor deve ser maior ou igual a zero; zero é a prioridade mais baixa. O seguinte valor especial também pode ser usado:

### **MQPRI\_PRIORITY\_AS\_Q\_DEF**

- Se a fila for uma fila de clusters, a prioridade para a mensagem será obtida do atributo **DefPriority** definido no gerenciador de filas de *destino* que possui a instância específica da fila na qual a mensagem é colocada.

Quando há múltiplas instâncias da fila de clusters e elas diferem nesse atributo, o valor de uma delas é escolhido e não pode ser previsto qual será usado. É necessário, portanto, configurar esse atributo para o mesmo valor em todas as instâncias. Se este não for o caso, a mensagem de erro AMQ9407 será emitida para os logs do gerenciador de filas. Consulte também [Como os atributos do objeto de destino são resolvidos para filas de alias, remotas e de clusters?](#)

O valor de *DefPriority* é copiado no campo *Priority* quando a mensagem é colocada na fila de destino. Se *DefPriority* for alterado subsequentemente, as mensagens que já foram colocadas na fila não serão afetadas

- Se a fila não for uma fila de clusters, a prioridade para a mensagem será obtida do atributo **DefPriority** definido no gerenciador de filas *local* , mesmo se o gerenciador de filas de destino for remoto.

Se houver mais de uma definição no caminho de resolução do nome da fila, a prioridade padrão será obtida do valor desse atributo na *primeira* definição no caminho. Ele pode ser:

- uma fila de alias
- Uma fila local
- Uma definição local de uma fila remota
- Um alias do gerenciador de filas
- Uma fila de transmissão (por exemplo, a fila *DefXmitQName* )

O valor de *DefPriority* é copiado no campo *Priority* quando a mensagem é colocada. Se *DefPriority* for alterado subsequentemente, as mensagens que já foram colocadas não serão afetadas

O valor retornado pela chamada MQGET é sempre maior ou igual a zero; o valor MQPRI\_PRIORITY\_AS\_Q\_DEF nunca é retornado.

Se uma mensagem for colocada com uma prioridade maior que o máximo suportado pelo gerenciador de filas locais (esse máximo é fornecido pelo atributo do gerenciador de filas **MaxPriority** ), a mensagem será aceita pelo gerenciador de filas, mas colocada na fila na prioridade máxima do gerenciador de filas; a chamada MQPUT ou MQPUT1 é concluída com MQCC\_WARNING e o código de razão MQRC\_PRIORITY\_EXCEEDS\_MAXIMUM. No entanto, o campo *Priority* retém o valor especificado pelo aplicativo que colocou a mensagem

No z/OS, se uma mensagem com um *MsgSeqNúmero* de 1 for colocada em uma fila que tenha uma seqüência de entrega de mensagens MQMDS\_PRIORITY e um tipo de índice MQIT\_GROUP\_ID, a fila poderá tratar a mensagem com uma prioridade diferente. Se a mensagem tiver sido colocada na fila com uma prioridade 0 ou 1, ela será processada como se tivesse uma prioridade 2. Isso ocorre porque a ordem das mensagens colocadas neste tipo de fila é otimizada para permitir testes de completude de grupo eficientes Para obter mais informações sobre a seqüência de entrega de mensagens MQMDS\_PRIORITY e o tipo de índice MQIT\_GROUP\_ID, consulte [MsgDelivery](#).

Ao responder a uma mensagem, os aplicativos devem usar a prioridade da mensagem de solicitação para a mensagem de resposta Em outras situações, especificar MQPRI\_PRIORITY\_AS\_Q\_DEF permite que o ajuste de prioridade seja executado sem alterar o aplicativo.

Esse é um campo de saída para a chamada MQGET e um campo de entrada para as chamadas MQPUT e MQPUT1 . O valor inicial desse campo é MQPRI\_PRIORITY\_AS\_Q\_DEF.

## **Persistência (MQLONG) para MQMD**

Isso indica se a mensagem sobrevive a falhas e reinicializações do sistema do gerenciador de filas. Para as chamadas MQPUT e MQPUT1, o valor deve ser um dos seguintes:

### **MQPER\_PERSISTENT**

A mensagem sobrevive a falhas e reinicializações do sistema do gerenciador de fila. Quando a mensagem tiver sido colocada e a unidade de trabalho na qual ela foi colocada tiver sido confirmada (se a mensagem for colocada como parte de uma unidade de trabalho), a mensagem será preservada no armazenamento auxiliar. Ela permanece lá até que a mensagem seja removida da fila e a unidade de trabalho na qual ela foi obtida foi confirmada (se a mensagem for recuperada como parte de uma unidade de trabalho).

Quando uma mensagem persistente é enviada para uma fila remota, um mecanismo de armazenamento e encaminhamento mantém a mensagem em cada gerenciador de fila ao longo da rota para o destino, até que a mensagem seja conhecida por ter chegado ao próximo gerenciador de fila.

Mensagens persistentes não podem ser colocadas em:

- Filas dinâmicas temporárias
- Filas compartilhadas que mapeiam para um objeto CFSTRUCT em CFLEVEL (2) ou abaixo ou onde o objeto CFSTRUCT é definido como RECOVER (NO).

Mensagens persistentes podem ser colocadas em filas dinâmicas permanentes e filas predefinidas.

### **MQPER\_NOT\_PERSISTENT**

A mensagem geralmente não sobrevive a falhas do sistema ou reinícios do gerenciador de filas. Isso se aplicará mesmo se uma cópia intacta da mensagem for localizada no armazenamento auxiliar quando o gerenciador de filas for reiniciado.

No caso de filas NPMCLASS (HIGH), mensagens não persistentes sobrevivem a um encerramento e reinicialização normais do gerenciador de filas.

No caso de filas compartilhadas, as mensagens não persistentes sobrevivem a reinicializações do gerenciador de fila no grupo de filas compartilhadas, mas não sobrevivem a falhas do recurso de acoplamento usado para armazenar mensagens nas filas compartilhadas.

### **MQPER\_PERSISTENCE\_AS\_Q\_DEF**

- Se a fila for uma fila de clusters, a persistência da mensagem será obtida do atributo **DefPersistence** definido no gerenciador de filas de *destino* que possui a instância específica da fila na qual a mensagem é colocada.

Quando há múltiplas instâncias da fila de clusters e elas diferem nesse atributo, o valor de uma delas é escolhido e não pode ser previsto qual será usado. É necessário, portanto, configurar esse atributo para o mesmo valor em todas as instâncias. Se este não for o caso, a mensagem de erro AMQ9407 será emitida para os logs do gerenciador de filas. Consulte também [Como os atributos do objeto de destino são resolvidos para filas de alias, remotas e de clusters?](#)

O valor de *DefPersistence* é copiado no campo *Persistence* quando a mensagem é colocada na fila de destino. Se *DefPersistence* for alterado subsequentemente, as mensagens que já foram colocadas na fila não serão afetadas.

- Se a fila não for uma fila de clusters, a persistência da mensagem será obtida do atributo **DefPersistence** definido no gerenciador de filas *local*, mesmo se o gerenciador de filas de destino for remoto.

Se houver mais de uma definição no caminho de resolução do nome da fila, a persistência padrão será obtida do valor desse atributo na *primeira* definição no caminho. Ele pode ser:

- uma fila de alias
- Uma fila local
- Uma definição local de uma fila remota



- Um alias do gerenciador de filas
- Uma fila de transmissão (por exemplo, a fila *DefXmitQName* )

O valor de *DefPersistence* é copiado no campo *Persistence* quando a mensagem é colocada. Se *DefPersistence* for alterado subsequentemente, as mensagens que já foram colocadas não serão afetadas

As mensagens persistentes e não persistentes podem existir na mesma fila.

Ao responder a uma mensagem, os aplicativos devem usar a persistência da mensagem de solicitação para a mensagem de resposta

Para uma chamada MQGET, o valor retornado é MQPER\_PERSISTENT ou MQPER\_NOT\_PERSISTENT

Esse é um campo de saída para a chamada MQGET e um campo de entrada para as chamadas MQPUT e MQPUT1 . O valor inicial desse campo é MQPER\_PERSISTENCE\_AS\_Q\_DEF

### **MsgId (MQBYTE24) para MQMD**

Esta é uma cadeia de bytes utilizada para distinguir uma mensagem de outra. Geralmente, duas mensagens não devem ter o mesmo identificador de mensagem, embora isso não seja desaprovado pelo gerenciador de filas. O identificador de mensagem é uma propriedade permanente da mensagem e persiste em reinicializações do gerenciador de filas. Como o identificador de mensagem é uma cadeia de bytes e não uma cadeia de caracteres, o identificador de mensagens não é convertido entre conjuntos de caractere quando a mensagem flui de um gerenciador de filas para outro...

Para as chamadas MQPUT e MQPUT1 , se MQMI\_NONE ou MQPMO\_NEW\_MSG\_ID for especificado pelo aplicativo, o gerenciador de filas gerará um identificador de mensagem exclusivo<sup>3</sup> quando a mensagem é colocada e a coloca no descritor de mensagens enviado com a mensagem. O gerenciador de filas também retorna esse identificador de mensagens no descritor de mensagem pertencente ao aplicativo de envio. O aplicativo pode usar esse valor para registrar informações sobre mensagens específicas e responder a consultas de outras partes do aplicativo.

Se a mensagem estiver sendo colocada em um tópico, o gerenciador de filas gerará identificadores de mensagens exclusivos conforme necessário para cada mensagem publicada. Se MQPMO\_NEW\_MSG\_ID for especificado pelo aplicativo, o gerenciador de filas gera um identificador de mensagem exclusivo para retornar na saída. Se MQMI\_NONE for especificado pelo aplicativo, o valor do campo *MsgId* no MQMD ficará inalterado no retorno da chamadas.

Consulte a descrição de MQPMO\_RETAIN em [“Opções \(MQLONG\) para MQPMO”](#) na página 523 para obter mais detalhes sobre publicações retidas.

Se a mensagem estiver sendo colocada em uma lista de distribuição, o gerenciador de filas gerará identificadores de mensagens exclusivos conforme necessário, mas o valor do campo *MsgId* no MQMD não será alterado no retorno da chamada, mesmo se MQMI\_NONE ou MQPMO\_NEW\_MSG\_ID foi especificado. Se o aplicativo precisar saber os identificadores de mensagens gerados pelo gerenciador de filas, o aplicativo deverá fornecer registros MQPMR que contenham o campo *MsgId*

O aplicativo de envio também pode especificar um valor para o identificador de mensagem diferente de MQMI\_NONE; isso para o gerenciador de fila gerando um identificador de mensagem exclusivo. Um

<sup>3</sup> Um *MsgId* gerado pelo gerenciador de filas consiste em um identificador de produto de 4 bytes (AMQ- ou CSQ- em ASCII ou EBCDIC, em que - representa um caractere em branco), seguido por uma implementação específica do produto de uma sequência exclusiva. Em IBM MQ , isso contém os primeiros 12 caracteres do nome do gerenciador de fila e um valor derivado do relógio do sistema.. Todos os gerenciadores de filas que podem se intercomunicar devem, portanto, ter nomes diferentes nos primeiros 12 caracteres, para assegurar que os identificadores de mensagens sejam exclusivos. A capacidade de gerar uma sequência exclusiva também depende do relógio do sistema não ser alterado para trás. Para eliminar a possibilidade de um identificador de mensagem gerado pelo gerenciador de fila duplicar um gerado pelo aplicativo, o aplicativo deve evitar a geração de identificadores com caracteres iniciais no intervalo de A a I em ASCII ou EBCDIC (X'41 'até X'49' e X'C1'até X'C9'). Entretanto, o aplicativo não é impedido de gerar identificadores com caracteres iniciais nesses intervalos.

aplicativo que está encaminhando uma mensagem pode usar isso para propagar o identificador da mensagem original.

O gerenciador de filas não usa esse campo, exceto para:

- Gerar um valor exclusivo se solicitado, conforme descrito acima
- Entregar o valor para o aplicativo que emite a solicitação *get* para a mensagem
- Copie o valor para o campo *CorrelId* de qualquer mensagem de relatório que ele gera sobre essa mensagem (dependendo das opções *Report*)

Quando o gerenciador de fila ou um agente de canal de mensagens gera uma mensagem de relatório, ele configura o campo *MsgId* da maneira especificada pelo campo *Report* da mensagem original, MQRO\_NEW\_MSG\_ID ou MQRO\_PASS\_MSG\_ID. Os aplicativos que geram mensagens de relatório também devem fazer isso.

Para a chamada MQGET, *MsgId* é um dos cinco campos que podem ser usados para recuperar uma mensagem específica da fila. Normalmente, a chamada MQGET retorna a próxima mensagem na fila, mas uma mensagem específica pode ser obtida especificando um ou mais dos cinco critérios de seleção, em qualquer combinação; esses campos são:



- *MsgId*
- *CorrelId*
- *GroupId*
- *MsgSeqNumber*
- *Offset*

O aplicativo configura um ou mais desses campos para os valores necessários e, em seguida, configura as opções correspondentes MQMO\_\* no campo *MatchOptions* em MQGMO para usar esses campos como critérios de seleção. Apenas as mensagens que possuem os valores especificados nesses campos são candidatas para recuperação. O padrão para o campo *MatchOptions* (se não alterado pelo aplicativo) é corresponder ao identificador de mensagem e ao identificador de correlação.

No z/OS, os critérios de seleção que você pode usar são restritos pelo tipo de índice usado para a fila. Consulte o atributo da fila **IndexType** para obter detalhes adicionais.

Normalmente, a mensagem retornada é a *primeira* mensagem na fila que satisfaz os critérios de seleção. Mas se MQGMO\_BROWSE\_NEXT for especificado, a mensagem retornada será a *próxima* mensagem que satisfaz os critérios de seleção; a varredura para essa mensagem inicia com a mensagem *seguinte* a posição atual do cursor.

**Nota:** A fila é varrida sequencialmente para uma mensagem que satisfaça os critérios de seleção, portanto, os tempos de recuperação serão mais lentos do que se nenhum critério de seleção for especificado, especialmente se muitas mensagens precisarem ser varridas antes que uma adequada seja localizada. As exceções a isso são:

-  uma chamada MQGET por *CorrelId* em Multiplataformas de 64 bits, em que o índice *CorrelId* elimina a necessidade de executar uma varredura sequencial verdadeira
-  uma chamada MQGET por *IndexType* no z/OS.

Em ambos os casos, o desempenho da recuperação é melhorado.

Consulte [Tabela 495 na página 406](#) para obter mais informações sobre como os critérios de seleção são usados em várias situações.

Especificar MQMI\_NONE como o identificador de mensagem tem o mesmo efeito de não especificar MQMO\_MATCH\_MSG\_ID, ou seja, *qualquer* identificador de mensagem corresponde.

Esse campo será ignorado se a opção MQGMO\_MSG\_UNDER\_CURSOR for especificada no parâmetro **GetMsgOpts** na chamada MQGET.

No retorno de uma chamada MQGET, o campo *MsgId* é configurado para o identificador de mensagem da mensagem retornada (se houver).

O seguinte valor especial pode ser usado:

### **MQMI\_NONE**

Nenhum identificador de mensagem foi especificado.

O valor é binário zero para o comprimento do campo.

Para a linguagem de programação C, a constante MQMI\_NONE\_ARRAY também é definida; isso tem o mesmo valor que MQMI\_NONE, mas é uma matriz de caracteres em vez de uma sequência.

Este é um campo de entrada / saída para as chamadas MQGET, MQPUT e MQPUT1 . O comprimento desse campo é fornecido por MQ\_MSG\_ID\_LENGTH O valor inicial desse campo é MQMI\_NONE.

### **CorrelId (MQBYTE24) para MQMD**

O campo CorrelId é a propriedade no cabeçalho da mensagem que pode ser usada para identificar uma mensagem específica ou um grupo de mensagens

Esta é uma cadeia de bytes que o aplicativo pode utilizar para relacionar uma mensagem a outra, ou para relacionar a mensagem a outro trabalho que o aplicativo está executando. O identificador de correlação é uma propriedade permanente da mensagem e persiste em reinicializações do gerenciador de filas. Como o identificador de correlação é uma sequência de bytes e não uma sequência de caracteres, o identificador de correlação não é convertido entre conjuntos de caracteres quando a mensagem flui de um gerenciador de filas para outro.

Para chamadas MQPUT e MQPUT1 , o aplicativo pode especificar qualquer valor. O gerenciador de filas transmite esse valor com a mensagem e o entrega ao aplicativo que emite a solicitação get para a mensagem.

Se o aplicativo especificar MQPMO\_NEW\_CORREL\_ID, o gerenciador de filas gera um identificador de correlação exclusivo que é enviado com a mensagem e também retornado para o aplicativo de envio na saída da chamada MQPUT ou MQPUT1 .

Um identificador de correlações gerado pelo gerenciador de filas consiste em um identificador de produto de 3 bytes (AMQ ou CSQ em ASCII ou EBCDIC), seguido por um byte reservado e uma implementação específica do produto de uma sequência exclusiva. No IBM MQ essa sequência de implementação específica do produto contém os primeiros 12 caracteres do nome do gerenciador de filas e um valor derivado do relógio do sistema. Todos os gerenciadores de filas que podem se intercomunicar devem, portanto, ter nomes diferentes nos primeiros 12 caracteres para assegurar que os identificadores de mensagens sejam exclusivos. A capacidade de gerar uma sequência exclusiva também depende do relógio do sistema não ser alterado para trás. Para eliminar a possibilidade de um identificador de mensagem gerado pelo gerenciador de fila duplicar um gerado pelo aplicativo, o aplicativo deve evitar a geração de identificadores com caracteres iniciais no intervalo de A a I em ASCII ou EBCDIC (X'41 'até X'49' e X'C1'até X'C9'). Entretanto, o aplicativo não é impedido de gerar identificadores com caracteres iniciais nesses intervalos.

Esse identificador de correlação gerado é mantido com a mensagem, se ela for retida, e é usado como o identificador de correlação quando a mensagem é enviada como uma publicação para assinantes que especificam MQCI\_NONE no campo de ID SubCorrelno MQSD transmitido na chamada MQSUB. Consulte [Opções MQPMO](#) para obter mais detalhes sobre publicações retidas.

Quando o gerenciador de filas ou um agente do canal de mensagens gera uma mensagem de relatório, ele configura o campo *CorrelId* da forma especificada pelo campo *Report* da mensagem original, MQRO\_COPY\_MSG\_ID\_TO\_CORREL\_ID ou MQRO\_PASS\_CORREL\_ID. Os aplicativos que geram mensagens de relatório também devem fazer isso.

Para a chamada MQGET, *CorrelId* é um dos cinco campos que podem ser usados para selecionar uma mensagem específica a ser recuperada da fila. Consulte a descrição do campo *MsgId* para obter detalhes de como especificar valores para esse campo

Especificar MQCI\_NONE como identificador de correlação tem o mesmo efeito de não especificar MQMO\_MATCH\_CORREL\_ID, ou seja, *qualquer* identificador de correlação será correspondente.

Se a opção MQGMO\_MSG\_UNDER\_CURSOR for especificada no parâmetro **GetMsgOpts** na chamada MQGET, esse campo será ignorado

No retorno de uma chamada MQGET, o campo *CorrelId* é configurado para o identificador de correlação da mensagem retornada (se houver).

Os seguintes valores especiais podem ser usados:

#### **MQCI\_NONE**

Nenhum identificador de correlação é especificado.

O valor é binário zero para o comprimento do campo.

Para a linguagem de programação C, a constante MQCI\_NONE\_ARRAY também é definida; isso tem o mesmo valor que MQCI\_NONE, mas é uma matriz de caracteres em vez de uma sequência.

#### **MQCI\_NEW\_SESSION**

A mensagem é o início de uma nova sessão

Esse valor é reconhecido pelo CICS bridge como indicando o início de uma nova sessão, ou seja o início de uma nova sequência de mensagens.

Para a linguagem de programação C, a constante MQCI\_NEW\_SESSION\_ARRAY também é definida; ela tem o mesmo valor que MQCI\_NEW\_SESSION, mas é uma matriz de caracteres em vez de uma cadeia.

Para a chamada MQGET, este é um campo de entrada / saída.. Para as chamadas MQPUT e MQPUT1 , este é um campo de entrada se MQPMO\_NEW\_CORREL\_ID não for especificado, e um campo de saída se MQPMO\_NEW\_CORREL\_ID for especificado. O comprimento desse campo é fornecido por MQ\_CORREL\_ID\_LENGTH O valor inicial desse campo é MQCI\_NONE.

#### **Nota:**

Não é possível passar o identificador de correlação de uma publicação em uma hierarquia. O campo é usado pelo gerenciador de filas.

#### ***BackoutCount (MQLONG) para MQMD***

Esta é uma contagem do número de vezes que a mensagem foi retornada anteriormente pela chamada MQGET como parte de uma unidade de trabalho e subsequentemente restaurada. Ele ajuda o aplicativo a detectar erros de processamento baseados no conteúdo da mensagem. A contagem exclui chamadas MQGET que especificam quaisquer opções MQGMO\_BROWSE\_ \*.

A precisão dessa contagem é afetada pelo atributo da fila **HardenGetBackout** ; consulte [“Atributos para filas”](#) na página 865.

Em z/OS, um valor de 255 significa que a mensagem foi restaurada 255 ou mais vezes; o valor retornado nunca é maior que 255.

Este é um campo de saída para a chamada MQGET Ele é ignorado para as chamadas MQPUT e MQPUT1 . O valor inicial desse campo é 0.

#### ***ReplyToQ (MQCHAR48) para MQMD***

Este é o nome da fila de mensagens para a qual o aplicativo que emitiu a solicitação de obtenção para a mensagem envia mensagens MQMT\_REPLY e MQMT\_REPORT O nome é o nome local de uma fila definida no gerenciador de filas identificado por *ReplyToQMgr*. Essa fila não deve ser uma fila modelo, embora o gerenciador de filas de envio não verifique isso quando a mensagem é colocada

Para as chamadas MQPUT e MQPUT1 , esse campo não deve ficar em branco se o campo *MsgType* tiver o valor MQMT\_REQUEST ou se alguma mensagem de relatório for solicitada pelo campo *Report* . No entanto, o valor especificado (ou substituído) é transmitido para o aplicativo que emite a solicitação *get* para a mensagem, independentemente do tipo de mensagem.

Se o campo *ReplyToQMgr* estiver em branco, o gerenciador de filas locais consulta o nome *ReplyToQ* em suas definições de fila. Se existir uma definição local de uma fila remota com esse nome, o valor *ReplyToQ* na mensagem transmitida será substituído pelos valores do atributo **RemoteQName** da definição da fila remota e esse valor será retornado no descritor de mensagens quando o aplicativo de

recebimento emitir uma chamada MQGET para a mensagem. Se uma definição local de uma fila remota não existir, *ReplyToQ* permanecerá inalterado

Se o nome for especificado, ele poderá conter espaços em branco à direita; o primeiro caractere nulo e os caracteres que o seguem são tratados como espaços em branco. Caso contrário, não será feita nenhuma verificação de que o nome satisfaz as regras de nomenclatura para filas; isso também é verdadeiro para o nome transmitido, se o *ReplyToQ* for substituído na mensagem transmitida. A única verificação feita é que um nome foi especificado, se as circunstâncias o exigirem.

Se uma fila de resposta não for necessária, configure o campo *ReplyToQ* para espaços em branco ou (na linguagem de programação C) para a cadeia nula ou para um ou mais espaços em branco seguidos por um caractere nulo; não deixe o campo não inicializado.

Para a chamada MQGET, o gerenciador de filas sempre retorna o nome preenchido com espaços em branco no comprimento do campo.

Se uma mensagem que requer uma mensagem de relatório não puder ser entregue, e a mensagem de relatório também não puder ser entregue na fila especificada, a mensagem original e a mensagem de relatório vão para a fila de mensagens não entregues (não entregues) (consulte o atributo **DeadLetterQName** descrito em [“Atributos do gerenciador de filas”](#) na página 827).

Esse é um campo de saída para a chamada MQGET e um campo de entrada para as chamadas MQPUT e MQPUT1. O comprimento desse campo é fornecido por MQ\_Q\_NAME\_LENGTH. O valor inicial deste campo é a cadeia nula em C e 48 caracteres em branco em outras linguagens de programação.

### ***ReplyToQMgr (MQCHAR48) para MQMD***

Este é o nome do gerenciador de fila para o qual enviar a mensagem de resposta ou a mensagem de relatório. *ReplyToQ* é o nome local de uma fila definida neste gerenciador de filas.

Se o campo *ReplyToQMgr* estiver em branco, o gerenciador de filas locais consulta o nome *ReplyToQ* em suas definições de fila. Se existir uma definição local de uma fila remota com esse nome, o valor *ReplyToQMgr* na mensagem transmitida será substituído pelos valores do atributo **RemoteQMgrName** da definição da fila remota e esse valor será retornado no descritor de mensagens quando o aplicativo de recebimento emitir uma chamada MQGET para a mensagem. Se uma definição local de uma fila remota não existir, o *ReplyToQMgr* transmitido com a mensagem será o nome do gerenciador de filas locais.

Se o nome for especificado, ele poderá conter espaços em branco à direita; o primeiro caractere nulo e os caracteres que o seguem são tratados como espaços em branco. Caso contrário, não será feita nenhuma verificação de que o nome satisfaz as regras de nomenclatura para gerenciadores de filas ou de que esse nome é conhecido para o gerenciador de filas de envio; isso também é verdadeiro para o nome transmitido, se o *ReplyToQMgr* for substituído na mensagem transmitida.

Se uma fila de resposta não for necessária, configure o campo *ReplyToQMgr* para espaços em branco ou (na linguagem de programação C) para a cadeia nula ou para um ou mais espaços em branco seguidos por um caractere nulo; não deixe o campo não inicializado.

Para a chamada MQGET, o gerenciador de filas sempre retorna o nome preenchido com espaços em branco no comprimento do campo.

Esse é um campo de saída para a chamada MQGET e um campo de entrada para as chamadas MQPUT e MQPUT1. O comprimento desse campo é fornecido por MQ\_Q\_MGR\_NAME\_LENGTH. O valor inicial deste campo é a cadeia nula em C e 48 caracteres em branco em outras linguagens de programação.

### ***UserIdentifier (MQCHAR12) para MQMD***

Isso faz parte do **contexto de identidade** da mensagem. Para obter mais informações sobre o contexto da mensagem, consulte [“MQMD - descritor de mensagem”](#) na página 435 e [Contexto da mensagem](#).

*UserIdentifier* especifica o identificador de usuário do aplicativo que originou a mensagem. O gerenciador de filas trata essas informações como dados de caractere, mas não define o formato deles.

Após uma mensagem ser recebida, use *UserIdentifier* no campo *AlternateUserId* do parâmetro **ObjDesc** de uma chamada MQOPEN ou MQPUT1 subsequente para executar a verificação de autorização para o usuário *UserIdentifier* em vez de o aplicativo executar a abertura.

Quando o gerenciador de filas gera essas informações para uma chamada MQPUT ou MQPUT1 :

- No z/OS, o gerenciador de filas usa o *AlternateUserId* do parâmetro **ObjDesc** da chamada MQOPEN ou MQPUT1 se a opção MQOO\_ALTERNATE\_USER\_AUTHORITY ou MQPMO\_ALTERNATE\_USER\_AUTHORITY foi especificada. Se a opção relevante não foi especificada, o gerenciador de filas usa um identificador de usuário determinado a partir do ambiente.
- Em outros ambientes, o gerenciador de filas sempre usa um identificador de usuário determinado do ambiente.

Quando o identificador de usuários é determinado a partir do ambiente:

- No z/OS, o gerenciador de filas usa:
  - Para MVS (lote), o identificador de usuário do cartão JES JOB ou da tarefa iniciada
  - Para TSO, o identificador de usuário propagado para a tarefa durante o envio da tarefa
  - Para CICS, o identificador de usuários associado à tarefa
  - Para IMS, o identificador de usuários depende do tipo de aplicativo:
    - para:
      - Regiões BMP sem mensagem
      - Regiões IFP sem mensagem
      - BMP de mensagem e regiões IFP de mensagem que não emitiram uma chamada GU bem-sucedidao gerenciador de filas usa o identificador de usuário do cartão JES JOB da região ou o identificador de usuário TSO. Se estiverem em branco ou nulo, ele usará o nome do bloco de especificação de programa (PSB).
    - para:
      - BMP de mensagem e regiões IFP de mensagem que *emitiram* uma chamada GU bem-sucedida
      - Regiões MPPo gerenciador de filas usa um dos seguintes:
      - O identificador de usuário conectado associado à mensagem
      - O nome do terminal lógico (LTERM)
      - O identificador de usuários do cartão JES JOB da região
      - O identificador de usuário do TSO
      - O nome do PSB
- No IBM i, o gerenciador de filas usa o nome do perfil do usuário associado à tarefa do aplicativo
- No AIX and Linux, o gerenciador de filas usa:
  - O nome de logon do aplicativo
  - O identificador de usuário efetivo do processo se nenhum logon estiver disponível
  - O identificador de usuário associado à transação, se o aplicativo for uma transação CICS
- Em sistemas Windows , o gerenciador de filas usa os primeiros 12 caracteres do nome do usuário conectado.

Esse campo é normalmente um campo de saída gerado pelo gerenciador de filas, mas para uma chamada MQPUT ou MQPUT1 é possível tornar esse campo um campo de entrada / saída e especificar o campo UserIdentification em vez de permitir que o gerenciador de filas gere essas informações. Especifique MQPMO\_SET\_IDENTITY\_CONTEXT ou MQPMO\_SET\_ALL\_CONTEXT no



parâmetro `PutMsgOpts` e especifique um ID do usuário no campo `UserIdentifier` se não quiser que o gerenciador de filas gere o campo `UserIdentifier` para uma chamada `MQPUT` ou `MQPUT1`.

Para as chamadas `MQPUT` e `MQPUT1`, este é um campo de entrada / saída se `MQPMO_SET_IDENTITY_CONTEXT` ou `MQPMO_SET_ALL_CONTEXT` é especificado no parâmetro **`PutMsgOpts`**. Quaisquer informações após um caractere nulo no campo são descartadas. O gerenciador de filas converte o caractere nulo e quaisquer caracteres a seguir em brancos. Se `MQPMO_SET_IDENTITY_CONTEXT` ou `MQPMO_SET_ALL_CONTEXT` não for especificado, esse campo será ignorado na entrada e será um campo somente de saída.

Após a conclusão bem-sucedida de uma chamada `MQPUT` ou `MQPUT1`, esse campo contém o `UserIdentifier` que foi transmitido com a mensagem se ele foi colocado em uma fila. Esse será o valor de `UserIdentifier` que será mantido com a mensagem se ela for retida (consulte a descrição de `MQPMO_RETAIN` para obter mais detalhes sobre as publicações retidas), mas não será usado como o `UserIdentifier` quando a mensagem for enviada como uma publicação para os assinantes porque eles fornecem um valor para substituir `UserIdentifier` em todas as publicações enviadas para eles. Se a mensagem não tiver nenhum contexto, o campo estará inteiramente em branco.

Este é um campo de saída para a chamada `MQGET`. O comprimento desse campo é fornecido por `MQ_USER_ID_LENGTH`. O valor inicial deste campo é a sequência nula em C e 12 caracteres em branco em outras linguagens de programação.

### ***AccountingToken (MQBYTE32) para MQMD***

Este é o token de contabilidade, parte do *contexto de identidade* da mensagem.. Para obter mais informações sobre o contexto da mensagem, consulte [“MQMD - descritor de mensagem”](#) na página 435 e [Contexto da mensagem](#).

O `AccountingToken` permite que um aplicativo seja cobrado apropriadamente pelo trabalho feito como resultado da mensagem. O gerenciador de filas trata estas informações como uma sequência de bits e não verifica seu conteúdo.

O gerenciador de filas gera essas informações da seguinte forma:

- O primeiro byte do campo é configurado para o comprimento das informações de contabilidade presentes nos bytes a seguir; esse comprimento está no intervalo de zero a 30 e é armazenado no primeiro byte como um número inteiro binário.
- O segundo e os bytes subsequentes (conforme especificado pelo campo de comprimento) são configurados para as informações de contabilidade apropriadas para o ambiente.
  - **z/OS** No z/OS, as informações de contabilidade são configuradas como:
    - Para o lote z/OS, as informações de contabilidade do cartão JES JOB ou de uma instrução JES ACCT no cartão EXEC (separadores de vírgula são alterados para X'FF '). Essas informações são truncadas, se necessário, para 31 bytes
    - Para o TSO, o número da conta do usuário
    - Para CICS, o identificador da unidade de trabalho LU 6.2 (UEPUOWDS) (26 bytes).
    - Para IMS, o nome do PSB de 8 caracteres concatenado com o token de recuperação de 16 caracteres IMS
  - **IBM i** No IBM i, as informações de contabilidade são configuradas para o código de contabilidade para a tarefa
  - **Linux** **AIX** No AIX and Linux, as informações de contabilidade são configuradas para o identificador de usuário numérico, em caracteres ASCII.
  - **Windows** No Windows, as informações de contabilidade são configuradas para um identificador de segurança (SID) Windows em um formato compactado. O SID identifica exclusivamente o identificador de usuário armazenado no campo `UserIdentifier`. Quando o SID é armazenado no campo `AccountingToken`, a Autoridade do Identificador de 6 bytes (localizada no terceiro

e nos bytes subsequentes do SID) é omitida. Por exemplo, se o SID Windows tiver 28 bytes de comprimento, 22 bytes de informações do SID serão armazenados no campo *AccountingToken* .

- O último byte (byte 32) do campo de contabilidade é configurado para o tipo de token de contabilidade (neste caso MQACTT\_NT\_SECURITY\_ID, x'0b'):

#### **MQACTT\_CICS\_LUOW\_ID**

CICS identificador de LUOW

#### **Windows MQACTT\_NT\_SECURITY\_ID**

Identificador de segurança Windows .

#### **IBM i MQACTT\_OS400\_ACCOUNT\_TOKEN**

Token de contabilidade IBM i .

#### **UNIX MQACTT\_UNIX\_NUMERIC\_ID**

UNIX identificador numérico..

#### **MQACTT\_USER**

Token de contabilidade definido pelo usuário..

#### **MQACTT\_UNKNOWN**

Tipo de token contábil desconhecido

O tipo de token de contabilidade é configurado para um valor explícito apenas nos seguintes ambientes:

- **AIX** AIX
- **IBM i** IBM i
- **Linux** Linux
- **Windows** Windows

e para o IBM MQ MQI clients conectado a esses sistemas Em outros ambientes, o tipo de token de contabilidade é configurado para o valor MQACTT\_UNKNOWN. Nesses ambientes, use o campo *PutAppType* para deduzir o tipo de token de contabilidade recebido..

- Todos os outros bytes são configurados como zero binário.

Para as chamadas MQPUT e MQPUT1 , este é um campo de entrada / saída se MQPMO\_SET\_IDENTITY\_CONTEXT ou MQPMO\_SET\_ALL\_CONTEXT é especificado no parâmetro **PutMsgOpts** .. Se nem MQPMO\_SET\_IDENTITY\_CONTEXT nem MQPMO\_SET\_ALL\_CONTEXT for especificado, esse campo será ignorado na entrada e será um campo somente de saída.. Para obter mais informações sobre contexto da mensagem, consulte [Contexto da mensagem](#).

Após a conclusão bem-sucedida de uma chamada MQPUT ou MQPUT1 , esse campo contém o AccountingToken que foi transmitido com a mensagem se ele foi colocado em uma fila. Esse será o valor de AccountingToken que será mantido com a mensagem se ela for retida (consulte a descrição de MQPMO\_RETAIN em “Opções (MQLONG) para MQPMO” na página 523 para obter mais detalhes sobre publicações retidas), mas não será usado como o AccountingToken quando a mensagem for enviada como uma publicação para assinantes, pois eles fornecem um valor para substituir AccountingToken em todas as publicações enviadas para eles. Se a mensagem não tiver nenhum contexto, o campo será totalmente binário zero.

Este é um campo de saída para a chamada MQGET

Esse campo não está sujeito a nenhuma tradução baseada no conjunto de caracteres do gerenciador de filas; o campo é tratado como uma sequência de bits e não como uma sequência de caracteres.

O gerenciador de filas não faz nada com as informações neste campo O aplicativo deve interpretar as informações se ele desejar usar as informações para fins contábeis

É possível usar o seguinte valor especial para o campo AccountingToken :

#### **MQACT\_NONE**

Nenhum token de contabilidade especificado.



O valor é binário zero para o comprimento do campo.

Para a linguagem de programação C, a constante `MQACT_NONE_ARRAY` também é definida; ela tem o mesmo valor que `MQACT_NONE`, mas é uma matriz de caracteres em vez de uma sequência.

O comprimento desse campo é fornecido por `MQ_ACCOUNTING_TOKEN_LENGTH`. O valor inicial desse campo é `MQACT_NONE`.

### ***Dados de ApplIdentity(MQCHAR32) para MQMD***

Isso faz parte do **contexto de identidade** da mensagem. Para obter mais informações sobre o contexto da mensagem, consulte [“MQMD - descritor de mensagem”](#) na página 435 e [Contexto da mensagem](#).

*ApplIdentityData* são informações definidas pelo conjunto de aplicativos e podem ser usadas para fornecer informações adicionais sobre a mensagem ou seu originador. O gerenciador de filas trata essas informações como dados de caractere, mas não define o formato deles. Quando o gerenciador de filas gera essas informações, ele fica totalmente em branco.

Para as chamadas `MQPUT` e `MQPUT1`, este é um campo de entrada / saída se `MQPMO_SET_IDENTITY_CONTEXT` ou `MQPMO_SET_ALL_CONTEXT` é especificado no parâmetro **PutMsgOpts**. Se um caractere nulo estiver presente, o caractere nulo e quaisquer caracteres seguintes serão convertidos em espaços em branco pelo gerenciador de filas. Se nem `MQPMO_SET_IDENTITY_CONTEXT` nem `MQPMO_SET_ALL_CONTEXT` for especificado, esse campo será ignorado na entrada e será um campo somente de saída. Para obter mais informações sobre contexto da mensagem, consulte [Contexto da mensagem](#).

Após a conclusão bem-sucedida de uma chamada `MQPUT` ou `MQPUT1`, esse campo contém o *ApplIdentityData* que foi transmitido com a mensagem se ele foi colocado em uma fila. Esse será o valor de *ApplIdentityData* que será mantido com a mensagem se ela for retida (consulte a descrição de `MQPMO_RETAIN` para obter mais detalhes sobre as publicações retidas), mas não será usado como o *ApplIdentityData* quando a mensagem for enviada como uma publicação para os assinantes porque eles fornecem um valor para substituir *ApplIdentityData* em todas as publicações enviadas para eles. Se a mensagem não tiver nenhum contexto, o campo estará inteiramente em branco.

Este é um campo de saída para a chamada `MQGET`. O comprimento desse campo é fornecido por `MQ_APPL_IDENTITY_DATA_LENGTH`. O valor inicial desse campo é a sequência nula em C e 32 caracteres em branco em outras linguagens de programação.

### ***PutApplTipo (MQLONG) para MQMD***

Esse é o tipo de aplicativo que coloca a mensagem e faz parte do **contexto de origem** da mensagem. Para obter mais informações sobre o contexto da mensagem, consulte [“MQMD - descritor de mensagem”](#) na página 435 e [Contexto da mensagem](#).

*PutApplType* pode ter um dos seguintes tipos padrão. Também é possível definir seus próprios tipos, mas apenas com valores no intervalo `MQAT_USER_FIRST` até `MQAT_USER_LAST`.

#### **MQAT\_AIX**

AIX (mesmo valor que `MQAT_UNIX`).

#### **MQAT\_AMQP**

aplicativo do protocolo AMQP

#### **MQAT\_BROKER**

Broker.

#### **MQAT\_CICS**

Transação CICS.

#### **MQAT\_CICS\_BRIDGE**

CICS bridge.

#### **MQAT\_CICS\_VSE**

Transação CICS/VSE.

**MQAT\_DOS**

IBM MQ MQI client aplicação no PC DOS.

**MQAT\_DQM**

Agente do gerenciador de filas distribuído..

**MQAT\_GUARDIAN**

Aplicativo tandem Guardian (mesmo valor que MQAT\_NSK).

**MQAT\_IMS**

IMS .

**MQAT\_IMS\_BRIDGE**

Ponte do IMS

**MQAT\_JAVA**

Java.

**MQAT\_MVS**

Aplicativo MVS ou TSO (mesmo valor que MQAT\_ZOS).

**MQAT\_NOTES\_AGENT**

Lotus Notes Aplicativo do agente..

**MQAT\_OS390**

OS/390 (mesmo valor que MQAT\_ZOS).

**MQAT\_OS400**

IBM i .

**MQAT\_QMGR**

Gerenciador de Filas

**MQAT\_UNIX**

UNIX .

**MQAT\_VOS**

Aplicativo do Stratus VOS

**MQAT\_WINDOWS**

Aplicativo Windows de 16 bits

**MQAT\_WINDOWS\_NT**

aplicativo Windows de 32 bits.

**MQAT\_WLM**

Aplicativo do gerenciador de carga de trabalho do z/OS

**MQAT\_XCF**

XCF.

**MQAT\_ZOS**

z/OS .

**MQAT\_DEFAULT**

Tipo de aplicativo padrão..

Este é o tipo de aplicativo padrão para a plataforma na qual o aplicativo está em execução

**Nota:** O valor dessa constante é específico para o ambiente. Por isso, sempre compile o aplicativo usando os arquivos de cabeçalho, include ou COPY que são apropriados para a plataforma na qual o aplicativo será executado.

**MQAT\_UNKNOWN**

Use esse valor para indicar que o tipo de aplicativo é desconhecido, mesmo que outras informações de contexto estejam presentes.

**MQAT\_USER\_FIRST**

Valor mais baixo para o tipo de aplicativo definido pelo usuário.

**MQAT\_USER\_LAST**

Valor mais alto para o tipo de aplicativo definido pelo usuário.

O seguinte valor especial também pode ocorrer:

### **MQAT\_NO\_CONTEXT**

Esse valor é configurado pelo gerenciador de filas quando uma mensagem é inserida sem contexto (ou seja, a opção de contexto MQPMO\_NO\_CONTEXT é especificada).

Quando uma mensagem é recuperada, o *PutApplType* pode ser testado para esse valor para decidir se a mensagem tem contexto (é recomendado que o *PutApplType* nunca seja configurado como MQAT\_NO\_CONTEXT, por um aplicativo usando MQPMO\_SET\_ALL\_CONTEXT, se qualquer um dos outros campos de contexto não estiver em branco)

Quando o gerenciador de filas gera essas informações como resultado de uma colocação de aplicativo, o campo é configurado para um valor determinado pelo ambiente. No IBM i, ele é configurado como MQAT\_OS400; o gerenciador de filas nunca usa MQAT\_CICS no IBM i

Para as chamadas MQPUT e MQPUT1, este é um campo de entrada e saída se MQPMO\_SET\_ALL\_CONTEXT for especificado no parâmetro **PutMsgOpts** .. Se MQPMO\_SET\_ALL\_CONTEXT não for especificado, esse campo será ignorado na entrada e será um campo somente de saída

Este é um campo de saída para a chamada MQGET O valor inicial desse campo é MQAT\_NO\_CONTEXT.

### **PutApplNome (MQCHAR28) para MQMD**

Este é o nome do aplicativo que coloca a mensagem, e faz parte do *contexto de origem* da mensagem O conteúdo difere entre plataformas e pode diferir entre liberações.

Para obter mais informações sobre o contexto da mensagem, consulte [“MQMD - descritor de mensagem”](#) na página 435 e [Contexto da mensagem](#).

Em IBM MQ 9.1.2, é possível especificar o nome do aplicativo em linguagens de programação adicionais. Consulte [Especificando o nome do aplicativo em linguagens de programação suportadas](#) para obter mais informações.

O formato de *PutApplName* depende do valor de *PutApplType* e pode mudar de uma liberação para outra. Mudanças são raras, mas acontecem se o ambiente mudar.

Quando o gerenciador de filas configura este campo (ou seja, para todas as opções, exceto MQPMO\_SET\_ALL\_CONTEXT), ele configura o campo para um valor que é determinado pelo ambiente:

- **z/OS** No z/OS, o gerenciador de filas usa:
  - Para o lote z/OS, o nome da tarefa de 8 caracteres do cartão JES JOB
  - Para TSO, o identificador de usuário do TSO de 7 caracteres
  - Para CICS, o applid de 8 caracteres, seguido pelo tranid de 4 caracteres
  - Para IMS, o identificador do sistema IMS de 8 caracteres, seguido pelo nome do PSB de 8 caracteres
  - Para XCF, o nome do grupo XCF de 8 caracteres, seguido pelo nome do membro XCF de 16 caracteres
  - Para uma mensagem gerada por um gerenciador de filas, os primeiros 28 caracteres do nome do gerenciador de filas
  - Para o enfileiramento distribuído sem CICS, o nome da tarefa de 8 caracteres do inicializador de canais seguido pelo nome de 8 caracteres do módulo colocado na fila de mensagens não entregues seguido por um identificador de tarefa de 8 caracteres.

Cada nome ou nomes são preenchidos à direita com espaços em branco, como qualquer espaço no restante do campo. Quando há mais de um nome, não há separador entre eles

- **Windows** Nos sistemas Windows, o gerenciador de filas usa os seguintes nomes:
  - Para um aplicativo CICS, o nome da transação CICS
  - Para um aplicativo nãoCICS, os 28 caracteres mais à direita do nome completo do executável
- **IBM i** No IBM i, o gerenciador de filas usa o nome completo da tarefa..

- **Linux** **AIX** No AIX and Linux, o gerenciador de filas usa os nomes a seguir:
  - Para um aplicativo CICS , o nome da transação CICS
  - Para um aplicativo nãoCICS , o MQ solicita ao sistema operacional o nome do processo.. Isso é retornado como o nome do arquivo de programa, sem caminho completo Em seguida, o MQ coloca esse nome do processo no MQMD do MQMD.PutApplName conforme a seguir:

#### **AIX** **AIX**

Se o nome for menor ou igual a 28 bytes, então o nome será inserido, preenchido à direita com espaços.

Se o nome for maior que 28 bytes, os 28 bytes mais à esquerda do nome serão inseridos.

#### **Linux** **Linux**

Se o nome for menor ou igual a 15 bytes, o nome será inserido, preenchido com espaços à direita.

Se o nome for maior que 15 bytes, os 15 bytes mais à esquerda do nome serão inseridos, preenchidos à direita com espaços.

Por exemplo, se você executar `/opt/mqm/samp/bin/amqsput QNAME QMNAME`, o PutApplName será ' amqsput '. Há 21 caracteres de espaço de preenchimento neste campo MQCHAR28 . Observe que o caminho completo incluindo `/opt/mqm/samp/bin` não está incluído no Nome PutAppl.

Para as chamadas MQPUT e MQPUT1 , este é um campo de entrada e saída se MQPMO\_SET\_ALL\_CONTEXT for especificado no parâmetro **PutMsgOpts** .. Quaisquer informações após um caractere nulo no campo são descartadas. O caractere nulo e quaisquer caracteres a seguir são convertidos em branco pelo gerenciador de filas. Se MQPMO\_SET\_ALL\_CONTEXT não for especificado, esse campo será ignorado na entrada e será um campo somente de saída

### **PutDate (MQCHAR8) para MQMD**

Esta é a data em que a mensagem foi colocada e faz parte do **contexto de origem** da mensagem.. Para obter mais informações sobre o contexto da mensagem, consulte [“MQMD - descritor de mensagem”](#) na página 435 e [Contexto da mensagem](#).

O formato utilizado para a data em que esse campo é gerado pelo gerenciador de filas é:

- AAAAMMDD

em que os caracteres representam:

#### **AAAA**

ano (quatro dígitos numéricos)

#### **MM**

mês do ano (01 a 12)

#### **DD**

dia do mês (01 a 31)

O Tempo Médio de Greenwich (GMT) é usado para os campos *PutDate* e *PutTime* , sujeito ao relógio do sistema sendo configurado com precisão para GMT.

Se a mensagem foi colocada como parte de uma unidade de trabalho, a data é quando a mensagem foi colocada e não a data em que a unidade de trabalho foi confirmada.

Para as chamadas MQPUT e MQPUT1 , este é um campo de entrada e saída se MQPMO\_SET\_ALL\_CONTEXT for especificado no parâmetro **PutMsgOpts** .. O conteúdo do campo não é verificado pelo gerenciador de filas, exceto que qualquer informação após um caractere nulo no campo é descartada. O gerenciador de filas converte o caractere nulo e quaisquer caracteres a seguir em brancos. Se MQPMO\_SET\_ALL\_CONTEXT não for especificado, esse campo será ignorado na entrada e será um campo somente de saída

Este é um campo de saída para a chamada MQGET O comprimento desse campo é fornecido por MQ\_PUT\_DATE\_LENGTH O valor inicial deste campo é a cadeia nula em C e 8 caracteres em branco em outras linguagens de programações.

## ***PutTime (MQCHAR8) para MQMD***

Este é o horário em que a mensagem foi colocada e faz parte do **contexto de origem** da mensagem. Para obter mais informações sobre o contexto da mensagem, consulte [“MQMD - descritor de mensagem”](#) na página 435 e [Contexto da mensagem](#).

O formato usado para o horário em que esse campo é gerado pelo gerenciador de filas é:

- HHMMSSSTH

em que os caracteres representam (por ordem):

### **DOMS**

horas (00 a 23)

### **MM**

minutos (00 a 59)

### **SS**

segundos (00 a 59; ver nota)

### **T**

Décimos de segundo (0 a 9)

### **H**

centésimos de segundo (0 a 9)

**Nota:** Se o relógio do sistema for sincronizado com um padrão de tempo muito preciso, será possível em raras ocasiões que 60 ou 61 sejam retornados para os segundos em *PutTime*. Isso acontece quando os segundos bissextos são inseridos no padrão de tempo global.

O Tempo Médio de Greenwich (GMT) é usado para os campos *PutDate* e *PutTime*, sujeito ao relógio do sistema sendo configurado com precisão para GMT.

Se a mensagem foi colocada como parte de uma unidade de trabalho, o horário será quando a mensagem foi colocada e não o horário em que a unidade de trabalho foi confirmada.

Para as chamadas MQPUT e MQPUT1, este é um campo de entrada e saída se MQPMO\_SET\_ALL\_CONTEXT for especificado no parâmetro **PutMsgOpts**. O gerenciador de filas não verifica o conteúdo do campo, exceto que qualquer informação após um caractere nulo no campo é descartada. O gerenciador de filas converte o caractere nulo e quaisquer caracteres a seguir em brancos. Se MQPMO\_SET\_ALL\_CONTEXT não for especificado, esse campo será ignorado na entrada e será um campo somente de saída.

Este é um campo de saída para a chamada MQGET. O comprimento desse campo é fornecido por MQ\_PUT\_TIME\_LENGTH. O valor inicial deste campo é a cadeia nula em C e 8 caracteres em branco em outras linguagens de programações.

## ***AppOriginDados (MQCHAR4) para MQMD***

Isso faz parte do *contexto de origem* da mensagem.. Para obter mais informações sobre o contexto da mensagem, consulte [“MQMD - descritor de mensagem”](#) na página 435 e [Contexto da mensagem](#).

AppOriginData são informações definidas pelo conjunto de aplicativos que podem ser usadas para fornecer informações adicionais sobre a origem da mensagem. Por exemplo, ele poderia ser configurado por aplicativos em execução com autoridade de usuário adequada para indicar se os dados de identificação são confiáveis..

O gerenciador de filas trata essas informações como dados de caractere, mas não define o formato deles. Quando o gerenciador de filas gera essas informações, ele fica totalmente em branco.

Para as chamadas MQPUT e MQPUT1, este é um campo de entrada e saída se MQPMO\_SET\_ALL\_CONTEXT for especificado no parâmetro **PutMsgOpts**. Quaisquer informações após um caractere nulo no campo são descartadas. O gerenciador de filas converte o caractere nulo e quaisquer caracteres a seguir em brancos. Se MQPMO\_SET\_ALL\_CONTEXT não for especificado, esse campo será ignorado na entrada e será um campo somente de saída.

Este é um campo de saída para a chamada MQGET. O comprimento desse campo é fornecido por MQ\_APPL\_ORIGIN\_DATA\_LENGTH. O valor inicial desse campo é a sequência nula em C e 4 caracteres em branco em outras linguagens de programação.

Quando a mensagem é publicada, embora ApplOriginData esteja configurado, ela fica em branco na assinatura que ela recebe..

### **GroupId (MQBYTE24) para MQMD**

Esta é uma cadeia de bytes que é utilizada para identificar o grupo de mensagens específico ou a mensagem lógica à qual a mensagem física pertence. *GroupId* também será usado se a segmentação for permitida para a mensagem. Em todos esses casos, *GroupId* possui um valor não nulo e um ou mais dos sinalizadores a seguir é configurado no campo *MsgFlags* :

- MQMF\_MSG\_IN\_GROUP
- MQMF\_LAST\_MSG\_IN\_GROUP
- MQMF\_SEGMENT
- MQMF\_LAST\_SEGMENT
- MQMF\_SEGMENTATION\_ALLOWED

Se nenhuma dessas sinalizadores for configurada, *GroupId* terá o valor nulo especial MQGI\_NONE.

O aplicativo não precisa configurar esse campo na chamada MQPUT ou MQGET se:

- Na chamada MQPUT, MQPMO\_LOGICAL\_ORDER é especificado..
- Na chamada MQGET, MQMO\_MATCH\_GROUP\_ID não foi especificado.

Estas são as maneiras recomendadas de usar estas chamadas para mensagens que não são mensagens de relatório. No entanto, se o aplicativo requerer mais controle ou a chamada for MQPUT1, o aplicativo deverá assegurar que *GroupId* seja configurado para um valor apropriado.

Os grupos de mensagens e segmentos poderão ser processados corretamente apenas se o identificador de grupo for exclusivo... Por esse motivo, *aplicativos não devem gerar seus próprios identificadores de grupo* ; em vez disso, os aplicativos devem executar um dos seguintes procedimentos:

- Se MQPMO\_LOGICAL\_ORDER for especificado, o gerenciador de filas gerará automaticamente um identificador de grupo exclusivo para a primeira mensagem no grupo ou segmento da mensagem lógica e usará esse identificador de grupo para as mensagens restantes no grupo ou nos segmentos da mensagem lógica, portanto, o aplicativo não precisa executar nenhuma ação especial.. Este é o procedimento recomendado.
- Se MQPMO\_LOGICAL\_ORDER não for especificado, o aplicativo deverá solicitar que o gerenciador de filas gere o identificador de grupo, configurando *GroupId* como MQGI\_NONE na primeira chamada MQPUT ou MQPUT1 para uma mensagem no grupo ou segmento da mensagem lógica.. O identificador de grupo retornado pelo gerenciador de filas na saída dessa chamada deve então ser usado para as mensagens restantes no grupo ou segmentos da mensagem lógica. Se um grupo de mensagens contiver mensagens segmentados, o mesmo identificador de grupo deverá ser usado para todos os segmentos e mensagens no grupo..

Quando MQPMO\_LOGICAL\_ORDER não é especificado, mensagens em grupos e segmentos de mensagens lógicas podem ser colocadas em qualquer ordem (por exemplo, em ordem inversa), mas o identificador de grupo deve ser alocado pela chamada *first* MQPUT ou MQPUT1 que é emitida para qualquer uma dessas mensagens.

Na entrada para as chamadas MQPUT e MQPUT1 , o gerenciador de filas usa o valor descrito em Ordem física em uma fila. Na saída das chamadas MQPUT e MQPUT1 , o gerenciador de filas configura esse campo para o valor que foi enviado com a mensagem se o objeto aberto for uma única fila e não uma lista de distribuição, mas o deixa inalterado se o objeto aberto for uma lista de distribuição. No último caso, se o aplicativo precisar saber os identificadores de grupo gerados, o aplicativo deverá fornecer registros MQPMR contendo o campo *GroupId* ..

Na entrada para a chamada MQGET, o gerenciador de filas usa o valor descrito em [Tabela 495 na página 406](#). Na saída da chamada MQGET, o gerenciador de filas configura esse campo para o valor da mensagem recuperada.

O seguinte valor especial é definido:

### **MQGI\_NONE**

Nenhum identificador de grupo especificado.

O valor é binário zero para o comprimento do campo. Esse é o valor usado para mensagens que não estão em grupos, não segmentos de mensagens lógicas e para as quais a segmentação não é permitida.

Para a linguagem de programação C, a constante MQGI\_NONE\_ARRAY também é definida; ela possui o mesmo valor que MQGI\_NONE, mas é uma matriz de caracteres em vez de uma sequência.

O comprimento desse campo é fornecido por MQ\_GROUP\_ID\_LENGTH O valor inicial desse campo é MQGI\_NONE. Esse campo será ignorado se *Version* for menor que MQMD\_VERSION\_2.

### **MsgSeqNúmero (MQLONG) para MQMD**

Este é o número de sequência de uma mensagem lógica dentro de um grupo

Os números de sequência começam em 1 e aumentam em 1 para cada nova mensagem lógica no grupo, até um máximo de 999 999 999. Uma mensagem física que não esteja em um grupo possui um número de sequência 1.

O aplicativo não precisa configurar esse campo na chamada MQPUT ou MQGET se:

- Na chamada MQPUT, MQPMO\_LOGICAL\_ORDER é especificado..
- Na chamada MQGET, MQMO\_MATCH\_MSG\_SEQ\_NUMBER não é especificado..

Estas são as maneiras recomendadas de usar estas chamadas para mensagens que não são mensagens de relatório. No entanto, se o aplicativo requerer mais controle ou a chamada for MQPUT1, o aplicativo deverá assegurar que *MsgSeqNumber* seja configurado para um valor apropriado.

Na entrada para as chamadas MQPUT e MQPUT1, o gerenciador de filas usa o valor descrito em [Ordem física em uma fila](#). Na saída das chamadas MQPUT e MQPUT1, o gerenciador de filas configura esse campo para o valor que foi enviado com a mensagem..

Na entrada para a chamada MQGET, o gerente de filas usa o valor mostrado em [Tabela 495 na página 406](#). Na saída da chamada MQGET, o gerenciador de filas configura esse campo para o valor da mensagem recuperada.

O valor inicial deste campo é um. Esse campo será ignorado se *Version* for menor que MQMD\_VERSION\_2.

### **Deslocamento (MQLONG) para MQMD**

Este é o deslocamento em bytes dos dados na mensagem física a partir do início da mensagem lógica da qual os dados fazem parte. Esses dados são chamados *segmento*. O deslocamento está no intervalo de 0 a 999 999 999. Uma mensagem física que não é um segmento de uma mensagem lógica tem um deslocamento zero.

O aplicativo não precisa configurar esse campo na chamada MQPUT ou MQGET se:

- Na chamada MQPUT, MQPMO\_LOGICAL\_ORDER é especificado..
- Na chamada MQGET, MQMO\_MATCH\_OFFSET não é especificado..

Estas são as maneiras recomendadas de usar estas chamadas para mensagens que não são mensagens de relatório. No entanto, se o aplicativo não estiver em conformidade com essas condições ou a chamada for MQPUT1, o aplicativo deverá assegurar que *Offset* seja configurado para um valor apropriado.

Na entrada para as chamadas MQPUT e MQPUT1, o gerenciador de filas usa o valor descrito em [Ordem física em uma fila](#). Na saída das chamadas MQPUT e MQPUT1, o gerenciador de filas configura esse campo para o valor que foi enviado com a mensagem..



Para um relatório de mensagem de relatório sobre um segmento de uma mensagem lógica, o campo *OriginalLength* (desde que não seja MQOL\_UNDEFINED) é usado para atualizar o deslocamento nas informações do segmento retidas pelo gerenciador de filas..

Na entrada para a chamada MQGET, o gerente de filas usa o valor mostrado em [Tabela 495 na página 406](#). Na saída da chamada MQGET, o gerenciador de filas configura esse campo para o valor da mensagem recuperada.

O valor inicial deste campo é zero. Esse campo será ignorado se *Version* for menor que MQMD\_VERSION\_2.

### **MsgFlags (MQLONG) para MQMD**

MsgFlags são sinalizações que especificam atributos da mensagem ou controlam seu processamento.

MsgFlags são divididos nas seguintes categorias:

- Sinalizações de segmentação
- Sinalizações de status

**Sinalização de Segmentação:** Quando uma mensagem é muito grande para uma fila, uma tentativa de colocar a mensagem na fila geralmente falha. A segmentação é uma técnica pela qual o gerenciador de filas ou aplicativo divide a mensagem em partes menores chamadas segmentos e coloca cada segmento na fila como uma mensagem física separada. O aplicativo que recupera a mensagem pode recuperar os segmentos um por um ou solicitar que o gerenciador de filas remonte os segmentos em uma única mensagem retornada pela chamada MQGET. O último é obtido especificando a opção MQGMO\_COMPLETE\_MSG na chamada MQGET, e fornecendo um buffer que é grande o suficiente para acomodar a mensagem completa (Consulte [“MQGMO-Obter opções de mensagem” na página 379](#) para obter detalhes da opção MQGMO\_COMPLETE\_MSG.) Uma mensagem pode ser segmentado no gerenciador de filas de envio, em um gerenciador de fila intermediário ou no gerenciador de filas de destino..

É possível especificar um dos seguintes para controlar a segmentação de uma mensagem:

#### **MQMF\_SEGMENTATION\_INIBIDA**

Essa opção impede que a mensagem seja dividida em segmentos pelo gerenciador de filas. Se especificado para uma mensagem que já é um segmento, essa opção impede que o segmento seja dividido em segmentos menores.

O valor desse sinalizador é binário zero. Esse é o padrão.

#### **MQMF\_SEGMENTATION\_ALLOWED**

Esta opção permite que a mensagem seja dividida em segmentos pelo Gerenciador de Filas. Se especificado para uma mensagem que já é um segmento, esta opção permite que o segmento seja dividido em segmentos menores.. MQMF\_SEGMENTATION\_ALLOWED pode ser configurado sem MQMF\_SEGMENT ou MQMF\_LAST\_SEGMENT sendo configurado.

- No z/OS, o gerenciador de filas não suporta a segmentação de mensagens Se uma mensagem for muito grande para a fila, a chamada MQPUT ou MQPUT1 falha com o código de razão MQRC\_MSG\_TOO\_BIG\_FOR\_Q. No entanto, a opção MQMF\_SEGMENTATION\_ALLOWED ainda pode ser especificada e permite que a mensagem seja segmentada em um gerenciador de filas remotas

Quando o gerenciador de filas segmenta uma mensagem, o gerenciador de filas ativa a sinalização MQMF\_SEGMENT na cópia do MQMD que é enviado com cada segmento, mas não altera as configurações dessas sinalizações no MQMD fornecido pelo aplicativo na chamada MQPUT ou MQPUT1 . Para o último segmento na mensagem lógica, o gerenciador de filas também ativa a sinalização MQMF\_LAST\_SEGMENT no MQMD que é enviado com o segmento.

**Nota:** Tome cuidado ao colocar mensagens com MQMF\_SEGMENTATION\_ALLOWED mas sem MQPMO\_LOGICAL\_ORDER. Se a mensagem for:

- Não é um segmento, e
- Não em um grupo, e



- Não sendo encaminhado,

o aplicativo deve reconfigurar o campo *GroupId* para MQGI\_NONE antes de *cada chamada* MQPUT ou MQPUT1, para que o gerenciador de filas possa gerar um identificador de grupo exclusivo para cada mensagem. Se isso não for feito, as mensagens não relacionadas poderão ter o mesmo identificador de grupo, o que pode levar a um processamento incorreto subsequentemente. Consulte as descrições dos campos *GroupId* e MQPMO\_LOGICAL\_ORDER para obter mais informações sobre quando reconfigurar o campo *GroupId*.

O gerenciador de filas divide as mensagens em segmentos conforme necessário para que os segmentos (mais quaisquer dados de cabeçalho necessários) se ajustem na fila. No entanto, há um limite inferior para o tamanho de um segmento gerado pelo gerenciador de filas e apenas o último segmento criado a partir de uma mensagem pode ser menor que esse limite (o limite inferior para o tamanho de um segmento gerado pelo aplicativo é um byte). Os segmentos gerados pelo gerenciador de filas podem ter comprimento desigual. O gerenciador de filas processa a mensagem como segue:

- Os formatos definidos pelo usuário são divididos em limites que são múltiplos de 16 bytes; o gerenciador de filas não gera segmentos menores que 16 bytes (diferentes do último segmento).
- Formatos integrados diferentes de MQFMT\_STRING são divididos em pontos apropriados à natureza dos dados presentes. No entanto, o gerenciador de filas nunca divide uma mensagem no meio de uma estrutura de cabeçalho do IBM MQ. Isso significa que um segmento contendo uma estrutura de cabeçalho MQ única não pode ser dividido ainda mais pelo gerenciador de filas e, como resultado, o tamanho mínimo possível do segmento para essa mensagem é maior que 16 bytes.

O segundo segmento ou posterior gerado pelo gerenciador de filas começa com um dos seguintes:

- Uma estrutura do cabeçalho MQ
- O início dos dados da mensagem do aplicativo
- Parte do caminho através dos dados da mensagem do aplicativo
- MQFMT\_STRING é dividido sem considerar a natureza dos dados presentes (SBCS, DBCS ou SBCS/DBCS combinados). Quando a cadeia é DBCS ou SBCS/DBCS misto, isso pode resultar em segmentos que não podem ser convertidos de um conjunto de caracteres para outro.. O gerenciador de filas nunca divide mensagens MQFMT\_STRING em segmentos menores que 16 bytes (diferentes do último segmento).
- O gerenciador de filas configura os campos *Format*, *CodedCharSetId* e *Encoding* no MQMD de cada segmento para descrever corretamente os dados presentes no *início* do segmento; o nome do formato é o nome de um formato integrado ou o nome de um formato definido pelo usuário..
- O campo *Report* no MQMD de segmentos com *Offset* maior que zero é modificado. Para cada tipo de relatório, se a opção de relatório for MQRO\_\*\_WITH\_DATA, mas o segmento não puder conter nenhum dos primeiros 100 bytes de dados do usuário (ou seja, os dados após quaisquer estruturas de cabeçalho IBM MQ que possam estar presentes), a opção de relatório será alterada para MQRO\_\*

O gerenciador de filas segue as regras acima, mas, caso contrário, divide mensagens imprevisivelmente; não faça suposições sobre onde uma mensagem é dividida.

Para mensagens *persistentes*, o gerenciador de filas pode executar a segmentação somente dentro de uma unidade de trabalho:

- Se a chamada MQPUT ou MQPUT1 estiver operando em uma unidade de trabalho definida pelo usuário, essa unidade de trabalho será usada. Se a chamada falhar durante o processo de segmentação, o gerenciador de filas remove quaisquer segmentos que foram colocados na fila como resultado da chamada com falha. Entretanto, a falha não impede que a unidade de trabalho seja confirmada com êxito.
- Se a chamada estiver operando fora de uma unidade de trabalho definida pelo usuário e não houver nenhuma unidade de trabalho definida pelo usuário existente, o gerenciador de filas criará uma unidade de trabalho apenas pela duração da chamada. Se a chamada for bem-sucedida, o gerenciador de filas confirma a unidade de trabalho automaticamente. Se a chamada falhar, o gerenciador de filas volta a unidade de trabalho.

- Se a chamada estiver operando fora de uma unidade de trabalho definida pelo usuário, mas uma unidade de trabalho definida pelo usuário existir, o gerenciador de filas não poderá executar a segmentação. Se a mensagem não precisar de segmentação, a chamada ainda poderá ser bem-sucedida. Mas se a mensagem requer segmentação, a chamada falha com o código de razão MQRC\_UOW\_NOT\_AVAILABLE.

Para mensagens *não persistentes*, o gerenciador de filas não requer que uma unidade de trabalho esteja disponível para executar a segmentação.

Tome cuidado especial ao converter dados em mensagens que podem ser segmentados:

- Se o aplicativo de recebimento converte dados na chamada MQGET e especifica a opção MQGMO\_COMPLETE\_MSG, a saída de conversão de dados recebe a mensagem completa para a saída converter e o fato de que a mensagem foi segmentada é aparente para a saída.
- Se o aplicativo de recebimento recupera um segmento por vez, a saída de conversão de dados é chamada para converter um segmento por vez. Portanto, a saída deve converter os dados em um segmento independentemente dos dados em qualquer um dos outros segmentos.

Se a natureza dos dados na mensagem for tal que a segmentação arbitrária dos dados em limites de 16 bytes pode resultar em segmentos que não podem ser convertidos pela saída ou o formato é MQFMT\_STRING e o conjunto de caracteres é DBCS ou SBCS/DBCS combinados, o aplicativo de envio deve criar e colocar os segmentos, especificando MQMF\_SEGMENTATION\_INIBIDO para suprimir segmentação adicional. Dessa forma, o aplicativo de envio pode assegurar que cada segmento contenha informações suficientes para permitir que a saída de conversão de dados converta o segmento com êxito.

- Se a conversão do emissor for especificada para um agente do canal de mensagens (MCA) de envio, o MCA converterá apenas mensagens que não sejam segmentos de mensagens lógicas; o MCA nunca tentará converter mensagens que sejam segmentos.

Esse sinalizador é uma sinalização de entrada nas chamadas MQPUT e MQPUT1 e uma sinalização de saída na chamada MQGET. Na última chamada, o gerenciador de fila também ecoa o valor da sinalização para o campo *Segmentation* em MQGMO.

O valor inicial desse sinalizador é MQMF\_SEGMENTATION\_INHIBITED

**Sinalizadores de status:** Estes são sinalizadores que indicam se a mensagem física pertence a um grupo de mensagens, é um segmento de uma mensagem lógica, ambos ou nenhum. Um ou mais dos seguintes podem ser especificados na chamada MQPUT ou MQPUT1 ou retornados pela chamada MQGET:

#### **MQMF\_MSG\_IN\_GROUP**

A mensagem é um membro de um grupo

#### **MQMF\_LAST\_MSG\_IN\_GROUP**

A mensagem é a última mensagem lógica em um grupo

Se esse sinalizador for configurado, o gerenciador de filas ativará MQMF\_MSG\_IN\_GROUP na cópia do MQMD que é enviada com a mensagem, mas não altera as configurações desses sinalizadores no MQMD fornecido pelo aplicativo na chamada MQPUT ou MQPUT1.

É válido que um grupo consista em apenas uma mensagem lógica. Se este for o caso, MQMF\_LAST\_MSG\_IN\_GROUP será configurado, mas o campo *MsgSeqNumber* terá o valor um.

#### **MQMF\_SEGMENT**

A mensagem é um segmento de uma mensagem lógica

Quando MQMF\_SEGMENT é especificado sem MQMF\_LAST\_SEGMENT, o comprimento dos dados da mensagem do aplicativo no segmento (*excluindo* os comprimentos de quaisquer estruturas de cabeçalho IBM MQ que possam estar presentes) deve ser pelo menos um. Se o comprimento for zero, a chamada MQPUT ou MQPUT1 falhará com o código de razão MQRC\_SEGMENT\_LENGTH\_ZERO.

No z/OS, essa opção não será suportada se a mensagem estiver sendo colocada em uma fila que tenha um tipo de índice MQIT\_GROUP\_ID.

#### **MQMF\_LAST\_SEGMENT**

A mensagem é o último segmento de uma mensagem lógica

Se esse sinalizador for configurado, o gerenciador de filas ativará MQMF\_SEGMENT na cópia do MQMD que é enviado com a mensagem, mas não altera as configurações desses sinalizadores no MQMD fornecido pelo aplicativo na chamada MQPUT ou MQPUT1 .

Uma mensagem lógica pode consistir em apenas um segmento. Se sim, MQMF\_LAST\_SEGMENT será configurado, mas o campo *Offset* terá o valor zero.

Quando MQMF\_LAST\_SEGMENT é especificado, o comprimento dos dados da mensagem do aplicativo no segmento ( *excluindo* os comprimento de quaisquer estruturas de cabeçalho que possam estar presentes) pode ser zero.

No z/OS, essa opção não será suportada se a mensagem estiver sendo colocada em uma fila que tenha um tipo de índice MQIT\_GROUP\_ID.

O aplicativo deve assegurar que essas sinalizações sejam configuradas corretamente ao colocar mensagens. Se MQPMO\_LOGICAL\_ORDER for especificado, ou foi especificado na chamada MQPUT precedente para o identificador de filas, as configurações dos sinalizadores devem ser consistentes com as informações do grupo e do segmento retidas pelo gerenciador de filas para o identificador de filas. As seguintes condições se aplicam a chamadas MQPUT *sucessivas* para o identificador de fila quando MQPMO\_LOGICAL\_ORDER é especificado:

- Se não houver nenhum grupo atual ou mensagem lógica, todas essas sinalizações (e combinações delas) serão válidas.
- Quando MQMF\_MSG\_IN\_GROUP tiver sido especificado, ele deverá permanecer ligado até MQMF\_LAST\_MSG\_IN\_GROUP ser especificado. A chamada falha com o código de razão MQRC\_INCOMPLETE\_GROUP se esta condição não for satisfeita.
- Quando MQMF\_SEGMENT tiver sido especificado, ele deverá permanecer até que MQMF\_LAST\_SEGMENT seja especificado. A chamada falha com o código de razão MQRC\_INCOMPLETE\_MSG se esta condição não for satisfeita.
- Depois que MQMF\_SEGMENT tiver sido especificado sem MQMF\_MSG\_IN\_GROUP, MQMF\_MSG\_IN\_GROUP deverá permanecer *off* até que MQMF\_LAST\_SEGMENT tenha sido especificado. A chamada falha com o código de razão MQRC\_INCOMPLETE\_MSG se esta condição não for satisfeita.

Ordem física em uma fila mostra as combinações válidas das sinalizações e os valores usados para vários campos.

Esses sinalizadores são sinalizadores de entrada nas chamadas MQPUT e MQPUT1 e sinalizadores de saída na chamada MQGET. Na última chamada, o gerenciador de filas também ecoa os valores dos sinalizadores para os campos *GroupStatus* e *SegmentStatus* em MQGMO.

Não é possível usar Publicar/Assinar com mensagens segmentadas ou agrupadas.

**Sinalizadores padrão:** O seguinte pode ser especificado para indicar que a mensagem possui atributos padrão:

#### **MQMF\_NONE**

Nenhum sinalizador de mensagem (atributos de mensagem padrão).

Isso inibe a segmentação, e indica que a mensagem não está em um grupo e não é um segmento de uma mensagem lógica. O MQMF\_NONE é definido para auxiliar na documentação do programa. Não se pretende que esse sinalizador seja usado com qualquer outro, mas como seu valor é zero, tal uso não pode ser detectado.

O campo *MsgFlags* é particionado em subcampos; para obter mais detalhes, consulte [“Opções de relatório e sinalizações de mensagem”](#) na página 935.

O valor inicial desse campo é MQMF\_NONE.. Esse campo será ignorado se *Version* for menor que MQMD\_VERSION\_2.

#### **OriginalLength (MQLONG) para MQMD**

Esse campo é relevante apenas para mensagens de relatório que são segmentos. Especifica o comprimento do segmento de mensagem ao qual a mensagem de relatório está relacionada; não

especifica o comprimento da mensagem lógica da qual o segmento faz parte ou o comprimento dos dados na mensagem de relatório.

**Nota:** Ao gerar uma mensagem de relatório para uma mensagem que é um segmento, o gerenciador de filas e o agente do canal de mensagens copiam no MQMD para a mensagem de relatório os campos *GroupId*, *MsgSeqNumber*, *Offset* e *MsgFlags* da mensagem original. Como resultado, a mensagem de relatório também é um segmento. Os aplicativos que geram mensagens de relatório devem fazer o mesmo e configurar o campo *OriginalLength* corretamente.

O seguinte valor especial é definido:

#### **MQOL\_UNDEFINED**

Comprimento original da mensagem não definido.

*OriginalLength* é um campo de entrada nas chamadas MQPUT e MQPUT1, mas o valor que o aplicativo fornece é aceito somente em circunstâncias específicas:

- Se a mensagem que está sendo colocada for um segmento e também uma mensagem de relatório, o gerenciador de filas aceita o valor especificado. O valor deve ser:
  - Maior que zero se o segmento não for o último segmento
  - Não menor que zero se o segmento for o último segmento
  - Não inferior ao comprimento dos dados presentes na mensagem

Se essas condições não forem atendidas, a chamada falhará com o código de razão MQRC\_ORIGINAL\_LENGTH\_ERROR

- Se a mensagem que está sendo colocada for um segmento, mas não uma mensagem de relatório, o gerenciador de filas ignorará o campo e usará o comprimento dos dados da mensagem do aplicativo.
- Em todos os outros casos, o gerenciador de filas ignora o campo e usa o valor MQOL\_UNDEFINED.

Este é um campo de saída na chamada MQGET

O valor inicial desse campo é MQOL\_UNDEFINED. Esse campo será ignorado se *Version* for menor que MQMD\_VERSION\_2.

## **MQMDE-Extensão do descritor de mensagens**

A estrutura MQMDE descreve os dados que às vezes ocorrem antes dos dados da mensagem do aplicativo. A estrutura contém os campos MQMD que existem no MQMD version-2, mas não no MQMD version-1.

### **Disponibilidade**

Todos os sistemas IBM MQ, além de IBM MQ MQI clients conectados a esses sistemas

### **Nome do Formato**

MQFMT\_MD\_EXTENSION

### **Conjunto de caracteres e codificação**

Os dados em MQMDE devem estar no conjunto de caracteres e na codificação do gerenciador de filas locais; eles são fornecidos pelo atributo do gerenciador de filas **CodedCharSetId** e MQENC\_NATIVE para a linguagem de programação C.

Configure o conjunto de caracteres e a codificação do MQMDE nos campos *CodedCharSetId* e *Encoding* em:

- O MQMD (se a estrutura MQMDE estiver no início dos dados da mensagem) ou
- A estrutura do cabeçalho que precede a estrutura MQMDE (todos os outros casos).

Se o MQMDE não estiver no conjunto de caracteres e na codificação do gerenciador de fila, o MQMDE será aceito, mas não honrado, ou seja, o MQMDE será tratado como dados da mensagem

**Nota:** No Windows, os aplicativos compilados com o Micro Focus COBOL usam um valor de MQENC\_NATIVE diferente da codificação do gerenciador de filas. Embora os campos numéricos na estrutura MQMD no MQPUT, MQPUT1 e chamadas MQGET devam estar na codificação Micro Focus COBOL, os campos numéricos na estrutura MQMDE devem estar na codificação do gerenciador de filas. Este último é fornecido por MQENC\_NATIVE para a linguagem de programação C e tem o valor 546..

## Uso

Os aplicativos que usam um MQMD version-2 não encontrarão uma estrutura MQMDE. No entanto, aplicativos especializados e aplicativos que continuam a usar um MQMD version-1 podem encontrar um MQMDE em algumas situações. A estrutura MQMDE pode ocorrer nas seguintes circunstâncias:

- Especificado nas chamadas MQPUT e MQPUT1
- Retornado pela chamada MQGET
- Em mensagens em filas de transmissão

## MQMDE especificado em chamadas MQPUT e MQPUT1

Nas chamadas MQPUT e MQPUT1, se o aplicativo fornecer um MQMD version-1, o aplicativo pode opcionalmente prefixar os dados da mensagem com um MQMDE, configurando o campo *Format* em MQMD para MQFMT\_MD\_EXTENSION para indicar que um MQMDE está presente. Se o aplicativo não fornecer um MQMDE, o gerenciador de filas assume valores padrão para os campos no MQMDE. Os valores padrão que o gerenciador de filas usa são os mesmos valores iniciais para a estrutura; consulte Tabela 503 na página 490.

Se o aplicativo fornecer um version-2 MQMD e prefixar os dados da mensagem do aplicativo com um MQMDE, as estruturas serão processada conforme mostrado na tabela a seguir:

<i>Tabela 502. Ação do gerenciador de filas quando MQMDE especificado em MQPUT ou MQPUT1 para MQMDE</i>			
<b>MQMD Versão</b>	<b>Valores de campos version-2</b>	<b>Valores de campos correspondentes em MQMDE</b>	<b>Ação executada pelo gerenciador da fila</b>
1	-	Válidos	MQMDE é honrado
2	Padrão	Válidos	MQMDE é honrado
2	Não padrão	Válidos	MQMDE é tratado como dados da mensagem
1 ou 2	Qualquer	Inválido	A chamada falha com um código de razão apropriado..
1 ou 2	Qualquer	MQMDE está no conjunto de caracteres ou na codificação errada ou é uma versão não suportada	MQMDE é tratado como dados da mensagem
<b>Nota:</b> No z/OS, se o aplicativo especificar um MQMD version-1 com um MQMDE, o gerenciador de filas validará o MQMDE apenas se a fila possuir um <i>IndexType</i> de MQIT_GROUP_ID..			

Há um caso especial. Se o aplicativo usar um MQMD version-2 para colocar uma mensagem que é um segmento (ou seja, o sinalizador MQMF\_SEGMENT ou MQMF\_LAST\_SEGMENT está configurado) e o nome do formato no MQMD for MQFMT\_DEAD\_LETTER\_HEADER, o gerenciador de filas gerará uma estrutura MQMDE e a inserirá *entre* a estrutura MQDLH e os dados que a seguem No MQMD que o gerenciador de filas retém com a mensagem, os campos version-2 são configurados para seus valores padrão

Vários dos campos que existem no MQMD version-2 , mas não no MQMD version-1 são campos de entrada / saída em MQPUT e MQPUT1. No entanto, o gerenciador de filas não retorna nenhum valor nos campos equivalentes no MQMDE na saída das chamadas MQPUT e MQPUT1 ; se o aplicativo requer esses valores de saída, ele deve usar um version-2 MQMD.

## MQMDE retornado pela chamada MQGET

Na chamada MQGET, se o aplicativo fornecer um MQMD version-1 , o gerenciador de filas prefixa a mensagem retornada com um MQMDE, mas somente se um ou mais dos campos no MQMDE tiver um valor não padrão. O gerenciador de filas configura o campo *Format* no MQMD para o valor MQFMT\_MD\_EXTENSION para indicar que um MQMDE está presente..

Se o aplicativo fornecer um MQMDE no início do parâmetro **Buffer** , o MQMDE será ignorado No retorno da chamada MQGET, ele é substituído pelo MQMDE para a mensagem (se for necessário) ou sobrescrito pelos dados da mensagem do aplicativo (se o MQMDE não for necessário).

Se a chamada MQGET retornar um MQMDE, os dados no MQMDE geralmente estão no conjunto de caracteres e na codificação do gerenciador de filas No entanto, o MQMDE pode estar em algum outro conjunto de caracteres e codificação se:

- O MQMDE foi tratado como dados na chamada MQPUT ou MQPUT1 (consulte [Tabela 502 na página 489](#) para obter as circunstâncias que podem causar isso.
- A mensagem foi recebida de um gerenciador de filas remotas conectado por uma conexão TCP e o MCA (Message Channel Agent) de recebimento não foi configurado corretamente.

**Nota:** No Windows, aplicativos compilados com Micro Focus COBOL usam um valor de MQENC\_NATIVE que é diferente da codificação do gerenciador de filas (veja acima).

## MQMDE em mensagens em filas de transmissão

Mensagens em filas de transmissão são prefixadas com a estrutura MQXQH, que contém dentro dela um MQMD version-1 . Um MQMDE também pode estar presente, posicionado entre a estrutura MQXQH e os dados da mensagem do aplicativo, mas ele geralmente estará presente apenas se um ou mais dos campos no MQMDE tiver um valor não padrão..

Outras estruturas de cabeçalho do MQ também podem ocorrer entre a estrutura MQXQH e os dados da mensagem do aplicativo Por exemplo, quando o cabeçalho de devoluções MQDLH está presente e a mensagem não é um segmento, a ordem é:

- MQXQH (contendo um MQMD version-1 )
- MQMDE
- MQDLH
- dados da mensagem do aplicativo;

## Campos

**Nota:** Na tabela a seguir, os campos são agrupados por uso e não alfabeticamente. Os tópicos filhos seguem a mesma sequência.

<i>Tabela 503. Campos em MQMDE para MQMDE</i>		
<b>Nome e descrição do campo</b>	<b>Nome da constante</b>	<b>Valor inicial (se houver) da constante</b>
<u>StrucId</u> (identificador de estrutura)	MQMDE_STRUC_ID	'MDE↵'
<u>Versão</u> (número da versão da estrutura).	MQMDE_VERSION_2	2
<u>StrucLength</u> (comprimento da estrutura MQMDE)	MQMDE_LENGTH_2	72

Tabela 503. Campos em MQMDE para MQMDE (continuação)

Nome e descrição do campo	Nome da constante	Valor inicial (se houver) da constante
Codificação (codificação numérica de dados que seguem MQMDE)	MQENC_NATIVE	Depende do ambiente
CodedCharSetId (identificador do conjunto de caracteres de dados que segue MQMDE)..	MQCCSI_UNDEFINED	0
Formato (nome do formato de dados após MQMDE)	MQFMT_NONE	Espaços em branco
Flags (sinalizadores gerais)	MQMDEF_NONE	0
GroupId (identificador de grupo).	MQGI_NONE	Nulos
MsgSeqNúmero (número de sequência da mensagem lógica dentro do grupo)	Nenhum	1
Deslocamento (deslocamento de dados na mensagem física do início da mensagem lógica)	Nenhum	0
MsgFlags (sinalizações de mensagens)	MQMF_NONE	0
OriginalLength (comprimento da mensagem original)	MQOL_UNDEFINED	-1

**Notas:**

1. O símbolo ~ representa um único caractere em branco.
2. Na linguagem de programação C, a variável macroMQMDE\_DEFAULT contém os valores que estão listados na tabela. Ele pode ser usado da seguinte maneira para fornecer valores iniciais para os campos na estrutura:

```
MQMDE MyMDE = {MQMDE_DEFAULT};
```

## Declarações de idiomas

### Declaração C para MQMDE

```
typedef struct tagMQMDE MQMDE;
struct tagMQMDE {
    MQCHAR4   StrucId;           /* Structure identifier */
    MQLONG    Version;          /* Structure version number */
    MQLONG    StrucLength;      /* Length of MQMDE structure */
    MQLONG    Encoding;         /* Numeric encoding of data that follows
                                MQMDE */
    MQLONG    CodedCharSetId;   /* Character-set identifier of data that
                                follows MQMDE */
    MQCHAR8   Format;           /* Format name of data that follows
                                MQMDE */
    MQLONG    Flags;            /* General flags */
    MQBYTE24  GroupId;         /* Group identifier */
    MQLONG    MsgSeqNumber;     /* Sequence number of logical message
                                within group */
    MQLONG    Offset;           /* Offset of data in physical message from
                                start of logical message */
    MQLONG    MsgFlags;        /* Message flags */
    MQLONG    OriginalLength;   /* Length of original message */
};
```

### Declaração COBOL para MQMDE

```
** MQMDE structure
```

```

10 MQMDE.
** Structure identifier
15 MQMDE-STRUCID PIC X(4).
** Structure version number
15 MQMDE-VERSION PIC S9(9) BINARY.
** Length of MQMDE structure
15 MQMDE-STRUCLength PIC S9(9) BINARY.
** Numeric encoding of data that follows MQMDE
15 MQMDE-ENCODING PIC S9(9) BINARY.
** Character-set identifier of data that follows MQMDE
15 MQMDE-CODEDCHARSETID PIC S9(9) BINARY.
** Format name of data that follows MQMDE
15 MQMDE-FORMAT PIC X(8).
** General flags
15 MQMDE-FLAGS PIC S9(9) BINARY.
** Group identifier
15 MQMDE-GROUPID PIC X(24).
** Sequence number of logical message within group
15 MQMDE-MSGSEQNUMBER PIC S9(9) BINARY.
** Offset of data in physical message from start of logical message
15 MQMDE-OFFSET PIC S9(9) BINARY.
** Message flags
15 MQMDE-MSGFLAGS PIC S9(9) BINARY.
** Length of original message
15 MQMDE-ORIGINALLENGTH PIC S9(9) BINARY.

```

### Declaração PL/I para MQMDE

```

dcl
1 MQMDE based,
3 StrucId char(4), /* Structure identifier */
3 Version fixed bin(31), /* Structure version number */
3 StrucLength fixed bin(31), /* Length of MQMDE structure */
3 Encoding fixed bin(31), /* Numeric encoding of data that
follows MQMDE */
3 CodedCharSetId fixed bin(31), /* Character-set identifier of data
that follows MQMDE */
3 Format char(8), /* Format name of data that follows
MQMDE */
3 Flags fixed bin(31), /* General flags */
3 GroupId char(24), /* Group identifier */
3 MsgSeqNumber fixed bin(31), /* Sequence number of logical message
within group */
3 Offset fixed bin(31), /* Offset of data in physical message from
start of logical message */
3 MsgFlags fixed bin(31), /* Message flags */
3 OriginalLength fixed bin(31); /* Length of original message */

```

### Declaração High Level Assembler para MQMDE

```

MQMDE          DSECT
MQMDE_STRUCID  DS   CL4  Structure identifier
MQMDE_VERSION  DS   F    Structure version number
MQMDE_STRUCLength DS   F    Length of MQMDE structure
MQMDE_ENCODING DS   F    Numeric encoding of data that follows
*              MQMDE
MQMDE_CODEDCHARSETID DS   F    Character-set identifier of data that
*              follows MQMDE
MQMDE_FORMAT   DS   CL8  Format name of data that follows MQMDE
MQMDE_FLAGS    DS   F    General flags
MQMDE_GROUPID  DS   XL24 Group identifier
MQMDE_MSGSEQNUMBER DS   F    Sequence number of logical message
*              within group
MQMDE_OFFSET   DS   F    Offset of data in physical message from
*              start of logical message
MQMDE_MSGFLAGS DS   F    Message flags
MQMDE_ORIGINALLENGTH DS   F    Length of original message
*
MQMDE_LENGTH   EQU   *-MQMDE
               ORG   MQMDE
MQMDE_AREA     DS   CL(MQMDE_LENGTH)

```

### Declaração Visual Basic para MQMDE

```
Type MQMDE
```



StrucId	As String*4	'Structure identifier'
Version	As Long	'Structure version number'
StrucLength	As Long	'Length of MQMDE structure'
Encoding	As Long	'Numeric encoding of data that follows' 'MQMDE'
CodedCharSetId	As Long	'Character-set identifier of data that' 'follows MQMDE'
Format	As String*8	'Format name of data that follows MQMDE'
Flags	As Long	'General flags'
GroupId	As MQBYTE24	'Group identifier'
MsgSeqNumber	As Long	'Sequence number of logical message within' 'group'
Offset	As Long	'Offset of data in physical message from' 'start of logical message'
MsgFlags	As Long	'Message flags'
OriginalLength	As Long	'Length of original message'
End Type		

### **StrucId (MQCHAR4) para MQMDE**

Este é o identificador da estrutura da extensão do descritor de mensagens. Ele é sempre um campo de entrada. Seu valor é MQMDE\_STRUC\_ID.

O valor deve ser:

#### **MQMDE\_STRUC\_ID**

Identificador para a estrutura de extensão do descritor de mensagem

Para a linguagem de programação C, a constante MQMDE\_STRUC\_ID\_ARRAY também é definida. Este tem o mesmo valor que MQMDE\_STRUC\_ID, mas é uma matriz de caracteres em vez de uma cadeia...

### **Versão (MQLONG) para MQMDE**

Este é o número da versão da estrutura; o valor deve ser:

#### **MQMDE\_VERSION\_2**

Estrutura da extensão do descritor de mensagens Version-2

A constante a seguir especifica o número da versão atual:

#### **MQMDE\_CURRENT\_VERSION**

A versão atual da estrutura de extensão do descritor de mensagem

O valor inicial desse campo é MQMDE\_VERSION\_2.

### **StrucLength (MQLONG) para MQMDE**

Este é o comprimento da estrutura MQMDE; o seguinte valor é definido:

#### **MQMDE\_LENGTH\_2**

Comprimento da estrutura de extensão do descritor de mensagens version-2

O valor inicial desse campo é MQMDE\_LENGTH\_2.

### **Codificação (MQLONG) para MQMDE**

Isso especifica a codificação numérica dos dados que seguem a estrutura MQMDE; ela não se aplica a dados numéricos na própria estrutura MQMDE

Na chamada MQPUT ou MQPUT1, o aplicativo deve configurar esse campo para o valor apropriado aos dados. O gerenciador de filas não verifica se o campo é válido.. Consulte o campo *Encoding* descrito em [“MQMD - descritor de mensagem”](#) na página 435 para obter mais informações sobre as codificações de dados

O valor inicial desse campo é MQENC\_NATIVE.

### **CodedCharSetId (MQLONG) para MQMDE**

Isso especifica o identificador do conjunto de caracteres dos dados que seguem a estrutura MQMDE; ele não se aplica aos dados de caracteres na própria estrutura MQMDE

Na chamada MQPUT ou MQPUT1 , o aplicativo deve configurar esse campo para o valor apropriado aos dados. O gerenciador de filas não verifica se esse campo é válido O seguinte valor especial pode ser usado:

### **MQCCSI\_INHERIT**

Os dados de caractere nos dados *após* essa estrutura estão no mesmo conjunto de caracteres que essa estrutura

O gerenciador de filas altera esse valor na estrutura enviada na mensagem para o identificador do conjunto de caractere real da estrutura.. Desde que não ocorra nenhum erro, o valor MQCCSI\_INHERIT não será retornado pela chamada MQGET

MQCCSI\_INHERIT não poderá ser usado se o valor do campo *PutApplType* no MQMD for MQAT\_BROKER..

Esse valor é suportado nos seguintes ambientes:

-  AIX
-  IBM i
-  Linux
-  Windows

e para clientes IBM MQ conectados a esses sistemas.

O valor inicial desse campo é MQCCSI\_UNDEFINED.

### **Formato (MQCHAR8) para MQMDE**

Especifica o nome do formato dos dados que seguem a estrutura MQMDE.

Na chamada MQPUT ou MQPUT1 , o aplicativo deve configurar esse campo para o valor apropriado aos dados. O gerenciador de filas não verifica se esse campo é válido Consulte o campo *Format* descrito em “MQMD - descritor de mensagem” na página 435 para obter mais informações sobre os nomes de formato

O valor inicial desse campo é MQFMT\_NONE.

### **Sinalizadores (MQLONG) para MQMDE**

A seguinte sinalização pode ser especificada:

#### **MQMDEF\_NONE**

Sem bandeiras.

O valor inicial desse campo é MQMDEF\_NONE.

### **GroupId (MQBYTE24) para MQMDE**

Consulte o campo *GroupId* descrito em “MQMD - descritor de mensagem” na página 435.. O valor inicial desse campo é MQGI\_NONE.

### **MsgSeqNúmero (MQLONG) para MQMDE**

Consulte o campo *MsgSeqNumber* descrito em “MQMD - descritor de mensagem” na página 435.. O valor inicial desse campo é 1.

### **Deslocamento (MQLONG) para MQMDE**

Consulte o campo *Offset* descrito em “MQMD - descritor de mensagem” na página 435.. O valor inicial desse campo é 0.

### **MsgFlags (MQLONG) para MQMDE**

Consulte o campo *MsgFlags* descrito em “MQMD - descritor de mensagem” na página 435.. O valor inicial desse campo é MQMF\_NONE..

### **OriginalLength (MQLONG) para MQMDE**

Consulte o campo *OriginalLength* descrito em “MQMD - descritor de mensagem” na página 435.. O valor inicial desse campo é MQOL\_UNDEFINED.

### **MQMHBO-Identificador de mensagens para opções de buffer**

A estrutura MQMHBO permite que os aplicativos especifiquem opções que controlam como os buffers são produzidos a partir de manipulações de mensagem. A estrutura é um parâmetro de entrada na chamada MQMHBUF..

### **Conjunto de caracteres e codificação**

Os dados em MQMHBO devem estar no conjunto de caracteres do aplicativo e na codificação do aplicativo (MQENC\_NATIVE)..

### **Campos**

**Nota:** Na tabela a seguir, os campos são agrupados por uso e não alfabeticamente. Os tópicos filhos seguem a mesma sequência.

<i>Tabela 504. Campos em MQMHBO</i>		
<b>Nome e descrição do campo</b>	<b>Nome da constante</b>	<b>Valor inicial (se houver) da constante</b>
StrucId (identificador de estrutura)	MQMHBO_STRUC_ID	'MHBO'
Versão (número da versão da estrutura).	MQMHBO_VERSION_1	1
Opções (opções que controlam a ação MQMHBUF)	MQMHBO_PROPERTIES_I N_MQRFH2	

**Notas:**

1. O valor Sequência nula ou espaços em branco indica a sequência nula em C e caracteres em branco em outras linguagens de programação.
2. Na linguagem de programação C, a variável macro MQMHBO\_DEFAULT contém os valores listados na tabela. Use-o da seguinte maneira para fornecer valores iniciais para os campos na estrutura:

```
MQMHBO MyMHBO = {MQMHBO_DEFAULT};
```

### **Declarações de idiomas**

Declaração C para MQMHBO

```
typedef struct tagMQMHBO MQMHBO;
struct tagMQMHBO {
    MQCHAR4   StrucId;           /* Structure identifier */
    MQLONG    Version;          /* Structure version number */
    MQLONG    Options;          /* Options that control the action of
                                MQMHBUF */
};
```

Declaração COBOL para MQMHBO

```
** MQMHBO structure
```

```

10 MQMHBO.
**  Structure identifier
   15 MQMHBO-STRUCID          PIC X(4).
**  Structure version number
   15 MQMHBO-VERSION         PIC S9(9) BINARY.
**  Options that control the action of MQMHBUF
   15 MQMHBO-OPTIONS        PIC S9(9) BINARY.

```

### Declaração PL/I para MQMHBO

```

Dcl
  1 MQMHBO based,
  3 StrucId      char(4),          /* Structure identifier */
  3 Version      fixed bin(31),   /* Structure version number */
  3 Options      fixed bin(31),   /* Options that control the action
                                of MQMHBUF */

```

### Declaração do High Level Assembler para MQMHBO

```

MQMHBO          DSECT
MQMHBO_STRUCID  DS   CL4  Structure identifier
MQMHBO_VERSION DS   F    Structure version number
MQMHBO_OPTIONS DS   F    Options that control the
*                action of MQMHBUF
MQMHBO_LENGTH  EQU   *-MQMHBO
MQMHBO_AREA    DS   CL(MQMHBO_LENGTH)

```

### **StrucId (MQCHAR4) para MQMHBO**

Este é o identificador de estrutura da manipulação de mensagens para a estrutura de opções de buffer... Ele é sempre um campo de entrada Seu valor é MQMHBO\_STRUC\_ID.

O valor deve ser:.

#### **MQMHBO\_STRUC\_ID**

Identificador da manipulação de mensagens para a estrutura de opções de buffer.

Para a linguagem de programação C, a constante MQMHBO\_STRUC\_ID\_ARRAY também é definida. Ele tem o mesmo valor que MQMHBO\_STRUC\_ID, mas é uma matriz de caracteres em vez de uma cadeia.

### **Versão (MQLONG) para MQMHBO**

Identificador de mensagem para a estrutura de opções de buffer-campo Versão

Este é o número da versão da estrutura O valor deve ser:.

#### **MQMHBO\_VERSION\_1**

Número da versão para a manipulação de mensagens para a estrutura de opções de buffer

A constante a seguir especifica o número da versão atual:

#### **MQMHBO\_CURRENT\_VERSION**

Versão atual da manipulação de mensagem para a estrutura de opções de buffer

Isto é sempre um campo de entrada. O valor inicial desse campo é MQMHBO\_VERSION\_1.

### **Opções (MQLONG) para MQMHBO**

Identificador de mensagem para estrutura de opções de buffer-campo Opções

Essas opções controlam a ação de MQMHBUF.

Você deve especificar a seguinte opção:

#### **MQMHBO\_PROPERTIES\_IN\_MQRFH2**

Ao converter propriedades de uma manipulação de mensagem em um buffer, converta-as no formato MQRFH2 .

Opcionalmente, também é possível especificar a opção a seguir: Para especificar mais de uma opção, inclua os valores juntos (não inclua a mesma constante mais de uma vez) ou combine os valores usando a operação OR bit a bit (se a linguagem de programação suportar operações de bit).

### **MQMHB0\_DELETE\_PROPERTIES**

As propriedades incluídas no buffer são excluídas do identificador de mensagem. Se a chamada falhar nenhuma propriedade será excluída.

Isto é sempre um campo de entrada. O valor inicial desse campo é MQMHBO\_PROPERTIES\_IN\_MQRFH2.

## **MQOD-Descrição de objetos**

A estrutura MQOD é usada para especificar um objeto por nome.. A estrutura é um parâmetro de entrada / saída nas chamadas MQOPEN e MQPUT1 .

Os seguintes tipos de objeto são válidos:

- Fila ou lista de distribuição
- Lista de Nomes
- Definição de processo
- Gerenciador de filas
- Tópico

## **Disponibilidade**

Todos os sistemas IBM MQ , além de IBM MQ MQI clients conectados a esses sistemas

## **Versão**

A versão atual do MQOD é MQOD\_VERSION\_4.. Os aplicativos que você deseja portar entre vários ambientes devem assegurar que a versão necessária do MQOD seja suportada em todos os ambientes em questão Os campos que existem somente nas versões mais recentes da estrutura são identificados como tal nas descrições a seguir:

Os arquivos de cabeçalho, COPY e INCLUDE fornecidos para as linguagens de programação suportadas contêm a versão mais recente do MQOD suportada pelo ambiente, mas com o valor inicial do campo *Version* configurado como MQOD\_VERSION\_1. Para usar campos que não estão presentes na estrutura *version-1* , o aplicativo deve configurar o campo *Version* para o número da versão necessária.

Para abrir uma lista de distribuição, *Version* deve ser MQOD\_VERSION\_2 ou superior.

## **Conjunto de caracteres e codificação**

Os dados em MQOD devem estar no conjunto de caracteres fornecido pelo atributo do gerenciador de filas do **CodedCharSetId** e pela codificação do gerenciador de filas locais fornecido por MQENC\_NATIVE No entanto, se o aplicativo estiver em execução como um cliente MQI MQ , a estrutura deverá estar no conjunto de caracteres e na codificação do cliente.

## **Campos**

**Nota:** Na tabela a seguir, os campos são agrupados por uso e não alfabeticamente. Os tópicos filhos seguem a mesma sequência.

<b>Nome e descrição do campo</b>	<b>Nome da constante</b>	<b>Valor inicial (se houver) da constante</b>
<u>StrucId</u> (identificador de estrutura)	MQOD_STRUC_ID	'0D-11'
<u>Versão</u> (número da versão da estrutura).	MQOD_VERSION_1	1
<u>ObjectType</u> (tipo de objeto)	MQOT_Q	1

Nome e descrição do campo	Nome da constante	Valor inicial (se houver) da constante
<u>ObjectName</u> (nome do objeto)	Nenhum	Cadeia nula ou espaços em branco
<u>ObjectQMgrNome</u> (nome do gerenciador de fila de objetos)	Nenhum	Cadeia nula ou espaços em branco
<u>DynamicQName</u> (nome da fila dinâmica)	Nenhum	'CSQ.*' em z/OS ; 'AMQ.*' caso contrário
<u>AlternateUserId</u> (identificador de usuário alternativo);	Nenhum	Cadeia nula ou espaços em branco
<b>Nota:</b> Os campos restantes serão ignorados se <i>Version</i> for menor que MQOD_VERSION_2.		
<u>RecsPresent</u> (número de registros de objeto presentes)	Nenhum	0
<u>KnownDestCount</u> (número de filas locais abertas com êxito)	Nenhum	0
<u>UnknownDestCount</u> (número de filas remotas abertas com sucesso)	Nenhum	0
<u>InvalidDestCount</u> (número de filas que falharam ao abrir)	Nenhum	0
<u>ObjectRecDeslocamento</u> (deslocamento do primeiro registro de objeto do início de MQOD)	Nenhum	0
<u>ResponseRecOffset</u> (deslocamento do primeiro registro de resposta do início do MQOD)	Nenhum	0
<u>ObjectRecPtr</u> (endereço do primeiro registro de objeto).	Nenhum	Ponteiro nulo ou bytes nulos
<u>ResponseRecPtr</u> (endereço do primeiro registro de resposta);	Nenhum	Ponteiro nulo ou bytes nulos
<b>Nota:</b> Os campos restantes serão ignorados se <i>Version</i> for menor que MQOD_VERSION_3..		
<u>AlternateSecurityId</u> (identificador de segurança alternativo)	MQSID_NONE	Nulos
<u>ResolvedQName</u> (nome da fila resolvido).	Nenhum	Cadeia nula ou espaços em branco
<u>ResolvedQMgrNome</u> (nome do gerenciador de filas resolvido)	Nenhum	Cadeia nula ou espaços em branco
<b>Nota:</b> Os campos restantes serão ignorados se <i>Version</i> for menor que MQOD_VERSION_4..		
<u>ObjectString</u> (nome do objeto longo);	MQCHARV_DEFAULT	Conforme definido para MQCHARV
<u>SelectionString</u> (sequência de seleção)	MQCHARV_DEFAULT	Conforme definido para MQCHARV
<u>ResObjectString</u> (nome do objeto longo resolvido)	MQCHARV_DEFAULT	Conforme definido para MQCHARV
<u>ResolvedType</u> (tipo de objeto resolvido)	MQOT_NONE	0

Nome e descrição do campo	Nome da constante	Valor inicial (se houver) da constante
<b>Notas:</b>		
<ol style="list-style-type: none"> <li>1. O símbolo ~ representa um único caractere em branco.</li> <li>2. O valor Sequência nula ou espaços em branco indica a sequência nula em C e caracteres em branco em outras linguagens de programação.</li> <li>3. Na linguagem de programação C, a variável macroMQOD_DEFAULT contém os valores listados na tabela. Ele pode ser usado da seguinte maneira para fornecer valores iniciais para os campos na estrutura: <div style="background-color: #f0f0f0; padding: 5px; margin-top: 10px;"> <pre>MQOD MyOD = {MQOD_DEFAULT};</pre> </div> </li> </ol>		

## Declarações de idiomas

### Declaração C para MQOD

```
typedef struct tagMQOD MQOD;
struct tagMQOD {
    MQCHAR4    StrucId;           /* Structure identifier */
    MQLONG     Version;          /* Structure version number */
    MQLONG     ObjectType;       /* Object type */
    MQCHAR48   ObjectName;       /* Object name */
    MQCHAR48   ObjectQMgrName;   /* Object queue manager name */
    MQCHAR48   DynamicQName;     /* Dynamic queue name */
    MQCHAR12   AlternateUserId;  /* Alternate user identifier */
    /* Ver:1 */
    MQLONG     RecsPresent;       /* Number of object records present */
    MQLONG     KnownDestCount;   /* Number of local queues opened
    successfully */
    MQLONG     UnknownDestCount; /* Number of remote queues opened
    successfully */
    MQLONG     InvalidDestCount; /* Number of queues that failed to
    open */
    MQLONG     ObjectRecOffset;  /* Offset of first object record from
    start of MQOD */
    MQLONG     ResponseRecOffset; /* Offset of first response record
    from start of MQOD */
    MQPTR      ObjectRecPtr;     /* Address of first object record */
    MQPTR      ResponseRecPtr;   /* Address of first response record */
    /* Ver:2 */
    MQBYTE40   AlternateSecurityId; /* Alternate security identifier */
    MQCHAR48   ResolvedQName;     /* Resolved queue name */
    MQCHAR48   ResolvedQMgrName; /* Resolved queue manager name */
    /* Ver:3 */
    MQCHARV    ObjectString;      /* Object Long name */
    MQCHARV    SelectionString;   /* Message Selector */
    MQCHARV    ResObjectString;   /* Resolved Long object name*/
    MQLONG     ResolvedType       /* Alias queue resolved
    object type */
    /* Ver:4 */
};
```

### Declaração COBOL para MQOD

```
** MQOD structure
10 MQOD.
** Structure identifier
15 MQOD-STRUCID                PIC X(4).
** Structure version number
15 MQOD-VERSION                PIC S9(9) BINARY.
** Object type
15 MQOD-OBJECTTYPE            PIC S9(9) BINARY.
** Object name
15 MQOD-OBJECTNAME            PIC X(48).
** Object queue manager name
15 MQOD-OBJECTQMRNAME        PIC X(48).
** Dynamic queue name
```

```

15 MQOD-DYNAMICQNAME          PIC X(48).
** Alternate user identifier
15 MQOD-ALTERNATEUSERID      PIC X(12).
** Number of object records present
15 MQOD-RECSPRESENT          PIC S9(9) BINARY.
** Number of local queues opened successfully
15 MQOD-KNOWNDSTCOUNT      PIC S9(9) BINARY.
** Number of remote queues opened successfully
15 MQOD-UNKNOWNDSTCOUNT    PIC S9(9) BINARY.
** Number of queues that failed to open
15 MQOD-INVALIDDSTCOUNT    PIC S9(9) BINARY.
** Offset of first object record from start of MQOD
15 MQOD-OBJECTRECOFFSET     PIC S9(9) BINARY.
** Offset of first response record from start of MQOD
15 MQOD-RESPONSERECOFFSET   PIC S9(9) BINARY.
** Address of first object record
15 MQOD-OBJECTRECPT        POINTER.
** Address of first response record
15 MQOD-RESPONSERECPT      POINTER.
** Alternate security identifier
15 MQOD-ALTERNATESECURITYID PIC X(40).
** Resolved queue name
15 MQOD-RESOLVEDQNAME       PIC X(48).
** Resolved queue manager name
15 MQOD-RESOLVEDQMGRNAME    PIC X(48).
** Object Long name
15 MQOD-OBJECTSTRING        .
** Address of variable length string
20 MQOD-OBJECTSTRING-VSPTR  POINTER.
** Offset of variable length string
20 MQOD-OBJECTSTRING-VSOFFSET PIC S9(9) BINARY.
** size of buffer
20 MQOD-OBJECTSTRING-VSBUFSIZE PIC S9(9) BINARY.
** Length of variable length string
20 MQOD-OBJECTSTRING-VSLENGTH PIC S9(9) BINARY.
** CCSID of variable length string
20 MQOD-OBJECTSTRING-VSCCSID PIC S9(9) BINARY.
** Message Selector
15 MQOD-SELECTIONSTRING     .
** Address of variable length string
20 MQOD-SELECTIONSTRING-VSPTR POINTER.
** Offset of variable length string
20 MQOD-SELECTIONSTRING-VSOFFSET PIC S9(9) BINARY.
** size of buffer
20 MQOD-SELECTIONSTRING-VSBUFSIZE PIC S9(9) BINARY.
** Length of variable length string
20 MQOD-SELECTIONSTRING-VSLENGTH PIC S9(9) BINARY.
** CCSID of variable length string
20 MQOD-SELECTIONSTRING-VSCCSID PIC S9(9) BINARY.
** Resolved Long object name
15 MQOD-RESOBJECTSTRING     .
** Address of variable length string
20 MQOD-RESOBJECTSTRING-VSPTR POINTER.
** Offset of variable length string
20 MQOD-RESOBJECTSTRING-VSOFFSET PIC S9(9) BINARY.
** size of buffer
20 MQOD-RESOBJECTSTRING-VSBUFSIZE PIC S9(9) BINARY.
** Length of variable length string
20 MQOD-RESOBJECTSTRING-VSLENGTH PIC S9(9) BINARY.
** CCSID of variable length string
20 MQOD-RESOBJECTSTRING-VSCCSID PIC S9(9) BINARY.
** Alias queue resolved object type
15 MQOD-RESOLVEDTYPE        PIC S9(9) BINARY.

```

## Declaração PL/I para MQOD

```

dcl
  1 MQOD based,
    3 StructId          char(4),          /* Structure identifier */
    3 Version           fixed bin(31),    /* Structure version number */
    3 ObjectType        fixed bin(31),    /* Object type */
    3 ObjectName        char(48),         /* Object name */
    3 ObjectQMgrName    char(48),         /* Object queue manager name */
    3 DynamicQName      char(48),         /* Dynamic queue name */
    3 AlternateUserId    char(12),        /* Alternate user identifier */
    3 RecsPresent        fixed bin(31),   /* Number of object records
                                         present */
    3 KnownDestCount    fixed bin(31),   /* Number of local queues opened
                                         successfully */

```



```

3 UnknownDestCount    fixed bin(31), /* Number of remote queues opened
                    successfully */
3 InvalidDestCount    fixed bin(31), /* Number of queues that failed to
                    open */
3 ObjectRecOffset     fixed bin(31), /* Offset of first object record
                    from start of MQOD */
3 ResponseRecOffset   fixed bin(31), /* Offset of first response record
                    from start of MQOD */
3 ObjectRecPtr        pointer,      /* Address of first object record */
3 ResponseRecPtr      pointer,      /* Address of first response
                    record */
3 AlternateSecurityId char(40),     /* Alternate security identifier */
3 ResolvedQName       char(48),     /* Resolved queue name */
3 ResolvedQMgrName    char(48),     /* Resolved queue manager name */
3 ObjectString,       /* Object Long name */
5 VSPtr              pointer,      /* Address of variable length string */
5 VSOffset           fixed bin(31), /* Offset of variable length string */
5 VSBufSize          fixed bin(31), /* size of buffer */
5 VSLength           fixed bin(31), /* Length of variable length string */
5 VSCCSID            fixed bin(31), /* CCSID of variable length string */
3 SelectionString,   /* Message Selection */
5 VSPtr              pointer,      /* Address of variable length string */
5 VSOffset           fixed bin(31), /* Offset of variable length string */
5 VSBufSize          fixed bin(31), /* size of buffer */
5 VSLength           fixed bin(31), /* Length of variable length string */
5 VSCCSID            fixed bin(31), /* CCSID of variable length string */
3 ResObjectString,   /* Resolved Long object name */
5 VSPtr              pointer,      /* Address of variable length string */
5 VSOffset           fixed bin(31), /* Offset of variable length string */
5 VSBufSize          fixed bin(31), /* size of buffer */
5 VSLength           fixed bin(31), /* Length of variable length string */
5 VSCCSID            fixed bin(31), /* CCSID of variable length string */
3 ResolvedType       fixed bin(31); /* Alias queue resolved object type */

```

## Declaração do High Level Assembler para MQOD

```

MQOD                DSECT
MQOD_STRUCID        DS    CL4    Structure identifier
MQOD_VERSION        DS    F      Structure version number
MQOD_OBJECTTYPE     DS    F      Object type
MQOD_OBJECTNAME     DS    CL48   Object name
MQOD_OBJECTQMGRNAME DS    CL48   Object queue manager name
MQOD_DYNAMICQNAME   DS    CL48   Dynamic queue name
MQOD_ALTERNATEUSERID DS    CL12  Alternate user identifier
MQOD_RECSPRESENT    DS    F      Number of object records present
MQOD_KNOWNDSTCOUNT DS    F      Number of local queues opened
*                   successfully
MQOD_UNKNOWNDSTCOUNT DS    F    Number of remote queues opened
*                   successfully
MQOD_INVALIDDSTCOUNT DS    F    Number of queues that failed to
*                   open
MQOD_OBJECTRECOFFSET DS    F      Offset of first object record from
*                   start of MQOD
MQOD_RESPONSERECOFFSET DS    F    Offset of first response record
*                   from start of MQOD
MQOD_OBJECTRECPTTR  DS    F      Address of first object record
MQOD_RESPONSERECPTR DS    F      Address of first response record
MQOD_ALTERNATESECURITYID DS    XL40  Alternate security identifier
MQOD_RESOLVEDQNAME  DS    CL48   Resolved queue name
MQOD_RESOLVEDQMGRNAME DS    CL48   Resolved queue manager name
MQOD_OBJECTSTRING   DS    F      Object Long name
MQOD_OBJECTSTRING_VSPTR DS    F    Address of variable length string
MQOD_OBJECTSTRING_VSOFFSET DS    F    Offset of variable length string
MQOD_OBJECTSTRING_VSBUFSIZE DS    F    size of buffer
MQOD_OBJECTSTRING_VSLENGTH DS    F    Length of variable length string
MQOD_OBJECTSTRING_VSCCSID DS    F    CCSID of variable length string
MQOD_OBJECTSTRING_LENGTH EQU    *- MQOD_OBJECTSTRING
ORG    MQOD_OBJECTSTRING
MQOD_OBJECTSTRING_AREA DS    CL(MQOD_OBJECTSTRING_LENGTH)
*
MQOD_SELECTIONSTRING DS    F      Message Selector
MQOD_SELECTIONSTRING_VSPTR DS    F    Address of variable length string
MQOD_SELECTIONSTRING_VSOFFSET DS    F    Offset of variable length string
MQOD_SELECTIONSTRING_VSBUFSIZE DS    F    size of buffer
MQOD_SELECTIONSTRING_VSLENGTH DS    F    Length of variable length string
MQOD_SELECTIONSTRING_VSCCSID DS    F    CCSID of variable length string
MQOD_SELECTIONSTRING_LENGTH EQU    *- MQOD_SELECTIONSTRING
ORG    MQOD_SELECTIONSTRING
MQOD_SELECTIONSTRING_AREA DS    CL(MQOD_SELECTIONSTRING_LENGTH)

```

```

*
MQOD_RESOBJECTSTRING      DS   F   Resolved Long object name
MQOD_RESOBJECTSTRING_VSPTR DS   F   Address of variable length string
MQOD_RESOBJECTSTRING_VSOFFSET DS  F   Offset of variable length string
MQOD_RESOBJECTSTRING_VSBUFSIZE DS  F   size of buffer
MQOD_RESOBJECTSTRING_VSLENGTH DS   F   Length of variable length string
MQOD_RESOBJECTSTRING_VSCCSID DS   F   CCSID of variable length string
MQOD_RESOBJECTSTRING_LENGTH EQU  *- MQOD_RESOBJECTSTRING
                           ORG  MQOD_RESOBJECTSTRING
MQOD_RESOBJECTSTRING_AREA DS   CL(MQOD_RESOBJECTSTRING_LENGTH)
MQOD_RESOLVEDTYPE         DS   F   Alias queue object resolved type
*
MQOD_LENGTH               EQU  *-MQOD
                           ORG  MQOD
MQOD_AREA                 DS   CL(MQOD_LENGTH)

```

## Declaração Visual Basic para MQOD

```

Type MQOD
  StrucId          As String*4  'Structure identifier'
  Version          As Long      'Structure version number'
  ObjectType       As Long      'Object type'
  ObjectName      As String*48  'Object name'
  ObjectQMgrName  As String*48  'Object queue manager name'
  DynamicQName    As String*48  'Dynamic queue name'
  AlternateUserId  As String*12  'Alternate user identifier'
  RecsPresent     As Long      'Number of object records present'
  KnownDestCount  As Long      'Number of local queues opened'
                          'successfully'
  UnknownDestCount As Long      'Number of remote queues opened'
                          'successfully'
  InvalidDestCount As Long      'Number of queues that failed to'
                          'open'
  ObjectRecOffset As Long      'Offset of first object record from'
                          'start of MQOD'
  ResponseRecOffset As Long     'Offset of first response record'
                          'from start of MQOD'
  ObjectRecPtr    As MQPTR     'Address of first object record'
  ResponseRecPtr  As MQPTR     'Address of first response record'
  AlternateSecurityId As MQBYTE40 'Alternate security identifier'
  ResolvedQName   As String*48  'Resolved queue name'
  ResolvedQMgrName As String*48  'Resolved queue manager name'
End Type

```

### **StrucId (MQCHAR4) para MQOD**

Este é o identificador da estrutura do descritor de objeto. Ele é sempre um campo de entrada Seu valor é MQOD\_STRUC\_ID.

O valor deve ser:.

#### **MQOD\_STRUC\_ID**

Identificador para a estrutura do descritor de objetos.

Para a linguagem de programação C, a constante MQOD\_STRUC\_ID\_ARRAY também é definida. Ele tem o mesmo valor que MQOD\_STRUC\_ID, mas é uma matriz de caracteres em vez de uma sequência.

### **Versão (MQLONG) para MQOD**

Este é o número da versão da estrutura; o valor deve ser um dos seguintes:

#### **MQOD\_VERSION\_1**

Estrutura do descritor de objeto Version-1

#### **MQOD\_VERSION\_2**

Estrutura do descritor de objeto Version-2 .

#### **MQOD\_VERSION\_3**

Estrutura do descritor de objeto Version-3

#### **MQOD\_VERSION\_4**

Estrutura do descritor de objeto Version-4

Todas as versões são suportadas em todos os ambientes do IBM MQ V7.0

Os campos que existem apenas nas versões mais recentes da estrutura são identificados como tal nas descrições dos campos A constante a seguir especifica o número da versão atual:

#### **MQOD\_CURRENT\_VERSION**

A versão atual da estrutura do descritor de objetos

Isto é sempre um campo de entrada. O valor inicial desse campo é MQOD\_VERSION\_1.

#### **ObjectType (MQLONG) para MQOD**

O tipo de objeto que está sendo nomeado no descritor de objeto Os valores possíveis são:

##### **MQOT\_CLNTCONN\_CHANEXO L**

Canal de conexão do cliente. O nome do objeto está localizado no campo *ObjectName*

##### **MQOT\_Q**

Fila. O nome do objeto está localizado no campo *ObjectName*

##### **MQOT\_NAMELIST**

Lista de nomes O nome do objeto está localizado no campo *ObjectName*

##### **MQOT\_PROCESS**

process definition. O nome do objeto está localizado no campo *ObjectName*

##### **MQOT\_Q\_MGR**

Gerenciador de Filas O nome do objeto está localizado no campo *ObjectName*

##### **MQOT\_TOPIC**

:NONE. O nome completo do tópico pode ser construído de dois campos diferentes: *ObjectName* e *ObjectString*.

Para obter detalhes sobre como esses dois campos são utilizados, consulte [Combinando sequências de tópicos](#).

Isto é sempre um campo de entrada. O valor inicial desse campo é MQOT\_Q.

#### **ObjectName (MQCHAR48) para MQOD**

Este é o nome local do objeto, conforme definido no gerenciador de filas identificado por *ObjectQMgrName* O nome pode conter os seguintes caracteres:

- Caracteres alfabéticos maiúsculos (A a Z)
- Caracteres alfabéticos minúsculos (a a z)
- Dígitos numéricos (0 a 9)
- Ponto (.), barra (/), sublinhado (\_), porcentagem (%)

O nome não deve conter espaços em branco iniciais ou integrados, mas pode conter rastreios de espaço em branco. Use um caractere nulo para indicar o término de dados significativos no nome; o nulo e quaisquer caracteres que o seguem são tratados como espaços em branco. As restrições a seguir aplicam-se aos ambientes indicados:

- Em sistemas que usam EBCDIC Katakana, caracteres minúsculos não podem ser usados.
- No z/OS:
  - Evite nomes que começam ou terminam com um sublinhado; eles não podem ser processados pelas operações e painéis de controle.
  - O caractere de porcentagem tem um significado especial para RACF Se RACF for usado como o gerenciador de segurança externa, os nomes não deverão conter o percentual Se forem, esses nomes não serão incluídos em nenhuma verificação de segurança quando os perfis genéricos do RACF forem usados.
- No IBM i, os nomes contendo caracteres minúsculos, barra ou porcentagem devem ser colocados entre aspas quando especificados em comandos. Essas aspas não devem ser especificadas para nomes que ocorrem como campos em estruturas ou como parâmetros em chamadas..

O nome completo do tópic pode ser construído de dois campos diferentes: *ObjectName* e *ObjectString*. Para obter detalhes de como esses dois campos são utilizados, consulte [Combinação de sequências de tópicos](#)

Os seguintes pontos aplicam-se aos tipos de objeto indicados:

- Se *ObjectName* for o nome de uma fila modelo, o gerenciador de filas criará uma fila dinâmica com os atributos da fila modelo e retornará no campo *ObjectName* o nome da fila criada. Uma fila modelo pode ser especificada apenas na chamada MQOPEN; uma fila modelo não é válida na chamadas MQPUT1 .
- Se *ObjectName* for o nome de uma fila de alias com TARGTYPE (TOPIC), uma verificação de segurança será feita pela primeira vez na fila de alias nomeada; isso é normal quando filas de alias são usadas. Quando a verificação de segurança for concluída com êxito, a chamada MQOPEN continuará e se comportará como uma chamada MQOPEN em um MQOT\_TOPIC; isso inclui fazer uma verificação de segurança no objeto do tópic administrativo.
- Se *ObjectName* e *ObjectQMgrName* identificarem uma fila compartilhada de propriedade do grupo de filas compartilhadas ao qual o gerenciador de filas locais pertence, não deverá haver também uma definição de fila com o mesmo nome no gerenciador de filas locais Se houver tal definição (uma fila local, fila de alias, fila remota ou fila modelo), a chamada falhará com o código de razão MQRC\_OBJECT\_NOT\_UNIQUE.
- Se o objeto que está sendo aberto for uma lista de distribuição (ou seja, *RecsPresent* está presente e maior que zero), *ObjectName* deverá estar em branco ou a sequência nula. Se essa condição não for atendida, a chamada falhará com o código de razão MQRC\_OBJECT\_NAME\_ERROR
- Se *ObjectType* for MQOT\_Q\_MGR, regras especiais se aplicarão; neste caso, o nome deve estar totalmente em branco até o primeiro caractere nulo ou o final do campo...

Este é um campo de entrada / saída para a chamada MQOPEN quando *ObjectName* é o nome de uma fila modelo e um campo somente de entrada em todos os outros casos O comprimento desse campo é fornecido por MQ\_Q\_NAME\_LENGTH O valor inicial deste campo é a cadeia nula em C e 48 caracteres em branco em outras linguagens de programação.

### **ObjectQMgrName (MQCHAR48) para MQOD**

Este é o nome do gerenciador de filas no qual o objeto *ObjectName* está definido Os caracteres que são válidos no nome são os mesmos para *ObjectName* (consulte [“ObjectName \(MQCHAR48\) para MQOD”](#) na página 503 ). Um nome que está inteiramente em branco até o primeiro caractere nulo ou o final do campo denota o gerenciador de fila ao qual o aplicativo está conectado (o gerenciador de filas locais)...

Os seguintes pontos aplicam-se aos tipos de objeto indicados:

- Se *ObjectType* for MQOT\_TOPIC, MQOT\_NAMELIST, MQOT\_PROCESS ou MQOT\_Q\_MGR, *ObjectQMgrName* deverá estar em branco ou ser o nome do gerenciador de fila local
- Se *ObjectName* for o nome de uma fila modelo, o gerenciador de filas criará uma fila dinâmica com os atributos da fila modelo e retornará no campo *ObjectQMgrName* o nome do gerenciador de filas no qual a fila é criada; este é o nome do gerenciador de fila local Uma fila modelo pode ser especificada apenas na chamada MQOPEN; uma fila modelo não é válida na chamadas MQPUT1 .
- Se *ObjectName* for o nome de uma fila de clusters e *ObjectQMgrName* estiver em branco, o destino das mensagens enviadas usando o identificador de filas retornado pela chamada MQOPEN será escolhido pelo gerenciador de fila (ou saída de carga de trabalho do cluster, se um estiver instalado) conforme a seguir:
  - Se MQOO\_BIND\_ON\_OPEN for especificado, o gerenciador de filas selecionará uma instância específica da fila de clusters durante o processamento da chamada MQOPEN e todas as mensagens colocadas usando esse identificador de filas serão enviadas para essa instância
  - Se MQOO\_BIND\_NOT\_FIXED for especificado, o gerenciador de filas poderá escolher uma instância diferente da fila de destino (que reside em um gerenciador de filas diferente no cluster) para cada chamada MQPUT sucessiva que usa esse identificador de filas.

Se o aplicativo precisar enviar uma mensagem para uma instância *específica* de uma fila de clusters (ou seja, uma instância de filas que reside em um gerenciador de filas específico no cluster), o aplicativo deverá especificar o nome desse gerenciador de filas no campo *ObjectQMgrName* .. Isso força o gerenciador de fila local a enviar a mensagem para o gerenciador de fila de destino especificado

- Se *ObjectName* for o nome de uma fila compartilhada que é de propriedade do grupo de filas compartilhadas ao qual o gerenciador de filas locais pertence, *ObjectQMgrName* pode ser o nome do grupo de filas compartilhadas, o nome do gerenciador de filas locais ou em branco; a mensagem é colocada na mesma fila, independentemente de qual desses valores for especificado

Grupos de filas compartilhadas são suportados somente no z/OS.

- Se *ObjectName* for o nome de uma fila compartilhada pertencente a um grupo de filas compartilhadas remotas (ou seja, um grupo de filas compartilhadas ao qual o gerenciador de filas locais não pertence), *ObjectQMgrName* deverá ser o nome do grupo de filas compartilhadas. É possível usar o nome de um gerenciador de fila que pertence a esse grupo, mas isso pode atrasar a mensagem se esse gerenciador de filas específico não estiver disponível ao chegar ao grupo de filas compartilhadas.
- Se o objeto que está sendo aberto for uma lista de distribuição (ou seja, *RecsPresent* for maior que zero), *ObjectQMgrName* deverá estar em branco ou a sequência nula. Se essa condição não for atendida, a chamada falhará com o código de razão MQRC\_OBJECT\_Q\_MGR\_NAME\_ERROR

Este é um campo de entrada / saída para a chamada MQOPEN quando *ObjectName* é o nome de uma fila modelo e um campo somente de entrada em todos os outros casos O comprimento desse campo é fornecido por MQ\_Q\_MGR\_NAME\_LENGTH O valor inicial deste campo é a cadeia nula em C e 48 caracteres em branco em outras linguagens de programação.

### ***DynamicQName (MQCHAR48) para MQOD***

Esse é o nome de uma fila dinâmica que deve ser criada pela chamada de MQOPEN Isso é relevante apenas quando *ObjectName* especifica o nome de uma fila modelo; em todos os casos, *DynamicQName* é ignorado.

Os caracteres que são válidos no nome são os mesmos para *ObjectName*, exceto que um asterisco também é válido. Um nome que está em branco (ou no qual apenas espaços em branco ocorrem antes do primeiro caractere nulo) não será válido se *ObjectName* for o nome de uma fila modelo.

Se o último caractere não em branco no nome for um asterisco (\*), o gerenciador de filas substituirá o asterisco por uma cadeia de caracteres que garante que o nome gerado para a fila seja exclusivo no gerenciador de fila local. Para permitir um número suficiente de caracteres para isso, o asterisco é válido apenas nas posições de 1 a 33 Não deve haver caracteres diferentes de espaços em branco ou um caractere nulo após o asterisco.

É válido que o asterisco ocorra na primeira posição de caractere, nesse caso, o nome consiste exclusivamente nos caracteres gerados pelo gerenciador de filas.

No z/OS, não use um nome com o asterisco na posição do primeiro caractere, pois não pode haver verificações de segurança feitas em uma fila com um nome completo gerado automaticamente.

Este é um campo de entrada. O comprimento desse campo é fornecido por MQ\_Q\_NAME\_LENGTH O valor inicial desse campo é determinado pelo ambiente:

- No z/OS, o valor é 'CSQ.\*'.
- Em outras plataformas, o valor é 'AMQ.\*'.

O valor é uma sequência terminada em nulo em C e uma sequência preenchida em branco em outras linguagens de programação.

### ***AlternateUserId (MQCHAR12) para MQOD***

Se você especificar MQOO\_ALTERNATE\_USER\_AUTHORITY para a chamada MQOPEN ou MQPMO\_ALTERNATE\_USER\_AUTHORITY para a chamada MQPUT1, esse campo contém um identificador de usuário alternativo que é usado para verificar a autorização para a abertura, no lugar do identificador de usuário sob o qual o aplicativo está atualmente em execução. No entanto, algumas

verificações ainda são realizadas com o identificador de usuário atual (por exemplo, verificações de contexto).

Se MQOO\_ALTERNATE\_USER\_AUTHORITY ou MQPMO\_ALTERNATE\_USER\_AUTHORITY for especificado e esse campo estiver inteiramente em branco até o primeiro caractere nulo ou o final do campo, a abertura poderá ser bem-sucedida somente se nenhuma autorização do usuário for necessária para abrir esse objeto com as opções especificadas...

Se nem MQOO\_ALTERNATE\_USER\_AUTHORITY nem MQPMO\_ALTERNATE\_USER\_AUTHORITY for especificado, esse campo será ignorado

As seguintes diferenças existem nos ambientes indicados:

- No z/OS, apenas os primeiros 8 caracteres de *AlternateUserId* são usados para verificar a autorização para a abertura... No entanto, o identificador de usuário atual deve ser autorizado a especificar esse identificador de usuário alternativo específico; todos os 12 caracteres do identificador de usuário alternativo são usados para essa verificação... O identificador de usuário deve conter apenas caracteres permitidos pelo gerenciador de segurança externo..

Se *AlternateUserId* for especificado para uma fila, o valor poderá ser usado subsequentemente pelo gerenciador de fila quando as mensagens forem colocadas. Se as opções MQPMO\_\*\_CONTEXT especificadas na chamada MQPUT ou MQPUT1 fizerem o gerenciador de filas gerar as informações de contexto de identidade, o gerenciador de filas colocará o *AlternateUserId* no campo *UserIdentifier* no MQMD da mensagem, no lugar do identificador de usuário atual.

- Em outros ambientes, *AlternateUserId* é usado apenas para verificações de controle de acesso no objeto sendo aberto. Se o objeto for uma fila, o *AlternateUserId* não afetará o conteúdo do campo *UserIdentifier* no MQMD de mensagens enviadas usando esse identificador de fila

Este é um campo de entrada. O comprimento desse campo é fornecido por MQ\_USER\_ID\_LENGTH O valor inicial deste campo é a sequência nula em C e 12 caracteres em branco em outras linguagens de programação.

### ***RecsPresent (MQLONG) para MQOD***

Este é o número de registros de objeto MQOR que foram fornecidos pelo aplicativo Se esse número for maior que zero, indica que uma lista de distribuição está sendo aberta, com *RecsPresent* sendo o número de filas de destino na lista. Uma lista de distribuição pode conter apenas um destino.

O valor de *RecsPresent* não deve ser menor que zero e, se for maior que zero, *ObjectType* deve ser MQOT\_Q; a chamada falhará com o código de razão MQRC\_RECS\_PRESENT\_ERROR se essas condições não forem satisfeitas.

Em z/OS, esse campo deve ser zero.

Este é um campo de entrada. O valor inicial desse campo é 0. Esse campo será ignorado se *Version* for menor que MQOD\_VERSION\_2.

### ***KnownDestContagem (MQLONG) para MQOD***

Este é o número de filas na lista de distribuição que são resolvidas para as filas locais e que foram abertas com sucesso A contagem não inclui filas que são resolvidas para filas remotas (mesmo que uma fila de transmissão local seja usada inicialmente para armazenar a mensagem). Se estiver presente, esse campo também será configurado ao abrir uma única fila que não esteja em uma lista de distribuição

Esse é um campo de saída. O valor inicial desse campo é 0. Esse campo será ignorado se *Version* for menor que MQOD\_VERSION\_1.

### ***UnknownDest(MQLONG) para MQOD***

Este é o número de filas na lista de distribuição que são resolvidas para filas remotas e que foram abertas com êxito Se estiver presente, esse campo também será configurado ao abrir uma única fila que não esteja em uma lista de distribuição

Esse é um campo de saída. O valor inicial desse campo é 0. Esse campo será ignorado se *Version* for menor que MQOD\_VERSION\_1.

### ***InvalidDestContagem (MQLONG) para MQOD***

Este é o número de filas na lista de distribuição que não foram abertas com êxito. Se estiver presente, esse campo também será configurado ao abrir uma única fila que não esteja em uma lista de distribuição

**Nota:** Se presente, esse campo será configurado somente se o parâmetro **CompCode** na chamada MQOPEN ou MQPUT1 for MQCC\_OK ou MQCC\_WARNING; ele não será configurado se o parâmetro **CompCode** for MQCC\_FAILED.

Esse é um campo de saída. O valor inicial desse campo é 0. Esse campo será ignorado se *Version* for menor que MQOD\_VERSION\_1.

### ***Deslocamento do ObjectRec(MQLONG) para MQOD***

Este é o deslocamento em bytes do primeiro registro de objeto MQOR do início da estrutura MQOD. O deslocamento pode ser positivo ou negativo *ObjectRecOffset* é usado apenas quando uma lista de distribuições está sendo aberta O campo será ignorado se *RecsPresent* for zero.

Quando uma lista de distribuição está sendo aberta, uma matriz de um ou mais registros de objeto MQOR deve ser fornecida para especificar os nomes das fila de destino na lista de distribuição. Isso pode ser feito de uma de duas maneiras:

- Usando o campo de deslocamento *ObjectRecOffset*.

Nesse caso, o aplicativo deve declarar sua própria estrutura contendo um MQOD seguido pela matriz de registros MQOR (com quantos elementos de matriz forem necessários) e configurar *ObjectRecOffset* para o deslocamento do primeiro elemento na matriz a partir do início do MQOD. Assegure-se de que esse deslocamento esteja correto e tenha um valor que possa ser acomodado em um MQLONG (a linguagem de programação mais restritiva é COBOL, para o qual o intervalo válido é -999 999 999 999 a +999 999 999).

Use *ObjectRecOffset* para linguagens de programação que não suportem o tipo de dados do ponteiro ou que implementem o tipo de dados do ponteiro de uma maneira que não seja móvel para diferentes ambientes (por exemplo, a linguagem de programação COBOL)..

- Usando o campo do ponteiro *ObjectRecPtr*

Nesse caso, o aplicativo pode declarar a matriz de estruturas MQOR separadamente da estrutura MQOD e configurar *ObjectRecPtr* para o endereço da matriz.

Use *ObjectRecPtr* para linguagens de programação que suportem o tipo de dados do ponteiro de uma maneira que seja móvel para ambientes diferentes (por exemplo, a linguagem de programação C)

Qualquer técnica que você escolher, use um dos *ObjectRecOffset* e *ObjectRecPtr*; a chamada falha com o código de razão MQRC\_OBJECT\_RECORDS\_ERROR se ambos forem zero ou ambos forem não zero.

Este é um campo de entrada. O valor inicial desse campo é 0. Esse campo será ignorado se *Version* for menor que MQOD\_VERSION\_2.

### ***ResponseRecOffset (MQLONG) para MQOD***

Este é o deslocamento em bytes do primeiro registro de resposta MQRR do início da estrutura MQOD. O deslocamento pode ser positivo ou negativo *ResponseRecOffset* é usado apenas quando uma lista de distribuições está sendo aberta O campo será ignorado se *RecsPresent* for zero.

Quando uma lista de distribuição está sendo aberta, é possível fornecer uma matriz de um ou mais registros de resposta MQRR para identificar as filas que falharam ao abrir (campo *CompCode* em MQRR) e a razão para cada falha (campo *Reason* em MQRR). Os dados são retornados na matriz de registros de resposta na mesma ordem em que os nomes de filas ocorrem na matriz de registros de objetos O gerenciador de filas configura os registros de respostas apenas quando o resultado da chamada é combinado (ou seja, algumas filas foram abertas com êxito enquanto outras falharam ou todas falharam, mas por razões diferentes); o código de razão MQRC\_MULTIPLE\_REASON da chamada indica este caso..

Se o mesmo código de razão se aplicar a todas as filas, esse motivo será retornado no parâmetro **Reason** da chamada MQOPEN ou MQPUT1 e os registros de resposta não serão configurados. Os registros de resposta são opcionais, mas se forem fornecidos, deverá haver *RecsPresent* deles.

Os registros de resposta podem ser fornecidos da mesma forma que os registros do objeto, especificando um deslocamento em *ResponseRecOffset* ou especificando um endereço em *ResponseRecPtr*; Para obter detalhes sobre como fazer isso, consulte [“Deslocamento do ObjectRec\(MQLONG\) para MQOD” na página 507](#) No entanto, não mais de um de *ResponseRecOffset* e *ResponseRecPtr* pode ser usado; a chamada falhará com o código de razão MQRC\_RESPONSE\_RECORDS\_ERROR se ambos forem diferentes de zero

Para a chamada MQPUT1, esses registros de resposta são usados para retornar informações sobre erros que ocorrem quando a mensagem é enviada para as filas na lista de distribuição, bem como erros que ocorrem quando as filas são abertas. O código de conclusão e o código de razão da operação put para uma fila substituem aqueles da operação open para essa fila somente se o código de conclusão do último era MQCC\_OK ou MQCC\_WARNING.

Este é um campo de entrada. O valor inicial desse campo é 0. Esse campo será ignorado se *Version* for menor que MQOD\_VERSION\_2.

### **ObjectRecPtr (MQPTR) para MQOD**

Este é o endereço do primeiro registro de objeto MQOR.. *ObjectRecPtr* é usado apenas quando uma lista de distribuições está sendo aberta O campo será ignorado se *RecsPresent* for zero.

É possível usar *ObjectRecPtr* ou *ObjectRecOffset* para especificar os registros de objeto, mas não ambos; para obter a descrição do campo *ObjectRecOffset*, consulte [“Deslocamento do ObjectRec\(MQLONG\) para MQOD” na página 507](#) Se você não usar *ObjectRecPtr*, configure-o para o ponteiro nulo ou bytes nulos.

Este é um campo de entrada. O valor inicial deste campo é o ponteiro nulo nas linguagens de programação que suportam ponteiros e, caso contrário, uma cadeia de bytes all-null. Esse campo será ignorado se *Version* for menor que MQOD\_VERSION\_2.

**Nota:** Em plataformas em que a linguagem de programação não suporta o tipo de dados do ponteiro, este campo é declarado como uma cadeia de bytes do comprimento apropriado, sendo o valor inicial a cadeia de bytes all-null.

### **ResponseRecPtr (MQPTR) para MQOD**

Este é o endereço do primeiro registro de resposta do MQRR *ResponseRecPtr* é usado apenas quando uma lista de distribuições está sendo aberta O campo será ignorado se *RecsPresent* for zero.

Use *ResponseRecPtr* ou *ResponseRecOffset* para especificar os registros de resposta, mas não ambos; para obter detalhes, consulte [“ResponseRecOffset \(MQLONG\) para MQOD” na página 507](#) Se você não usar *ResponseRecPtr*, configure-o para o ponteiro nulo ou bytes nulos.

Este é um campo de entrada. O valor inicial deste campo é o ponteiro nulo nas linguagens de programação que suportam ponteiros e, caso contrário, uma cadeia de bytes all-null. Esse campo será ignorado se *Version* for menor que MQOD\_VERSION\_2.

**Nota:** Em plataformas em que a linguagem de programação não suporta o tipo de dados do ponteiro, este campo é declarado como uma cadeia de bytes do comprimento apropriado, sendo o valor inicial a cadeia de bytes all-null.

### **AlternateSecurityId (MQBYTE40) para MQOD**

Este é um identificador de segurança que é transmitido com o *AlternateUserId* para o serviço de autorização para permitir que verificações de autorização apropriadas sejam executadas. *AlternateSecurityId* é usado apenas se:

- MQOO\_ALTERNATE\_USER\_AUTHORITY é especificado na chamada MQOPEN ou
- MQPMO\_ALTERNATE\_USER\_AUTHORITY é especificado na chamada MQPUT1,



e o campo *AlternateUserId* não está totalmente em branco até o primeiro caractere nulo ou o final do campo..

No Windows, *AlternateSecurityId* pode ser usado para fornecer o identificador de segurança (SID) Windows que identifica exclusivamente o *AlternateUserId*. O SID para um usuário pode ser obtido do sistema Windows usando a chamada API do `LookupAccountName()` Windows .

No z/OS, esse campo é ignorado..

O campo *AlternateSecurityId* possui a seguinte estrutura:

- O primeiro byte é um número inteiro binário contendo o comprimento dos dados significativos que seguem; o valor exclui o próprio byte de comprimento. Se não houver nenhum identificador de segurança presente, o comprimento será zero.
- O segundo byte indica o tipo de identificador de segurança que está presente; os valores a seguir são possíveis:

**MQSIDT\_NT\_SECURITY\_ID**

Identificador de segurança Windows .

**MQSIDT\_NONE**

Não há identificador de segurança.

- O terceiro e os bytes subsequentes até o comprimento definido pelo primeiro byte contêm o próprio identificador de segurança.
- Os bytes restantes no campo são configurados como binários zero.

É possível usar o seguinte valor especial:

**MQSID\_NONE**

Nenhum identificador de segurança especificado.

O valor é binário zero para o comprimento do campo.

Para a linguagem de programação C, a constante `MQSID_NONE_ARRAY` também é definida; ela tem o mesmo valor que `MQSID_NONE`, mas é uma matriz de caracteres em vez de uma sequência.

Este é um campo de entrada. O comprimento desse campo é fornecido por `MQ_SECURITY_ID_LENGTH`. O valor inicial desse campo é `MQSID_NONE`. Esse campo será ignorado se *Version* for menor que `MQOD_VERSION_3`..

### ***ResolvedQName (MQCHAR48) para MQOD***

Este é o nome da fila de destino após o gerenciador de fila local resolver o nome.. O nome retornado é o nome de uma fila existente no gerenciador de filas identificado por *ResolvedQMgrName*.

Um valor não em branco será retornado apenas se o objeto for uma única fila aberta para procura, entrada ou saída (ou qualquer combinação). Se o objeto aberto for qualquer um dos seguintes, *ResolvedQName* será configurado como espaços em branco:

- Não uma fila
- Uma fila, mas não aberta para procura, entrada ou saída
- Uma lista de distribuição
- Uma fila de alias que faz referência a um objeto do tópico (consulte [ResObjectString](#) no lugar).
- Uma fila de alias que é resolvida para um objeto de tópico

Esse é um campo de saída. O comprimento desse campo é fornecido por `MQ_Q_NAME_LENGTH`. O valor inicial deste campo é a cadeia nula em C e 48 caracteres em branco em outras linguagens de programação. Esse campo será ignorado se *Version* for menor que `MQOD_VERSION_3`..

### ***ResolvedQMgrNome (MQCHAR48) para MQOD***

Esse é o nome do gerenciador de fila de destino após o gerenciador de fila local resolver o nome.. O nome retornado é o nome do gerenciador de filas que possui a fila identificada por *ResolvedQName*.. *ResolvedQMgrName* pode ser o nome do gerenciador de fila local..

Se *ResolvedQName* for uma fila compartilhada pertencente ao grupo de filas compartilhadas ao qual o gerenciador de filas locais pertence, *ResolvedQMgrName* será o nome do grupo de filas compartilhadas. Se a fila for de propriedade de algum outro grupo de filas compartilhadas, o *ResolvedQName* poderá ser o nome do grupo de filas compartilhadas ou o nome de um gerenciador de filas que é um membro do grupo de filas compartilhadas (a natureza do valor retornado é determinada pelas definições de fila que existem no gerenciador de fila local).

Um valor não em branco será retornado apenas se o objeto for uma única fila aberta para procura, entrada ou saída (ou qualquer combinação). Se o objeto aberto for qualquer um dos seguintes, *ResolvedQMgrName* será configurado como espaços em branco:

- Não uma fila
- Uma fila, mas não aberta para procura, entrada ou saída
- Uma fila de clusters com MQOO\_BIND\_NOT\_FIXED especificado (ou com MQOO\_BIND\_AS\_Q\_DEF em vigor quando o atributo da fila **DefBind** tiver o valor MQBND\_BIND\_NOT\_FIXED)
- Uma lista de distribuição

Esse é um campo de saída. O comprimento desse campo é fornecido por MQ\_Q\_NAME\_LENGTH O valor inicial deste campo é a cadeia nula em C e 48 caracteres em branco em outras linguagens de programação. Esse campo será ignorado se *Version* for menor que MQOD\_VERSION\_3..

### **ObjectString (MQCHARV) para MQOD**

O campo ObjectString especifica o nome do objeto longo

Isso especifica o nome do objeto longo a ser usado. Esse campo é referido apenas para determinados valores de *ObjectType* e é ignorado para todos os outros valores. Consulte a descrição de *ObjectType* para obter detalhes de quais valores indicam que esse campo é utilizado

Se *ObjectString* for especificado incorretamente, de acordo com a descrição de como usar a estrutura MQCHARV ou se exceder o comprimento máximo, a chamada falhará com o código de razão MQRC\_OBJECT\_STRING\_ERROR.

Este é um campo de entrada. Os valores iniciais dos campos nessa estrutura são os mesmos da estrutura MQCHARV.

O nome completo do tópico pode ser construído de dois campos diferentes: *ObjectName* e *ObjectString*. Para obter detalhes de como esses dois campos são utilizados, consulte [Combinação de sequências de tópicos](#)

### **SelectionString (MQCHARV) para MQOD**

Essa é a sequência usada para fornecer os critérios de seleção usados ao recuperar mensagens fora de uma fila

*SelectionString* não deve ser fornecido nos seguintes casos:

- Se *ObjectType* não for MQOT\_Q
- Se a fila que está sendo aberta não estiver sendo aberta usando uma das opções MQOO\_BROWSE ou MQOO\_INPUT\_\*

Se *SelectionString* for fornecido nesses casos, a chamada falhará com o código de razão MQRC\_SELECTOR\_INVALID\_FOR\_TYPE..

Se *SelectionString* for especificado incorretamente, de acordo com a descrição de como usar a estrutura “MQCHARV-Sequência de Comprimento Variável” na página 301 ou se exceder o comprimento máximo, a chamada falhará com o código de razão MQRC\_SELECTOR\_STRING\_ERROR. O comprimento máximo de *SelectionString* é MQ\_SELECTOR\_LENGTH..

O uso do *SelectionString* é descrito em [Seletores](#).

### **Sequência ResObject(MQCHARV) para MQOD**

O campo String ResObjecté o nome do objeto longo após o gerenciador de filas resolver o nome fornecido no campo *ObjectName* .

Esse campo é retornado apenas para tópicos e aliases de filas que fazem referência a um objeto do tópico

Se o nome do objeto longo for fornecido em *ObjectString* e nada for fornecido em *ObjectName*, o valor retornado nesse campo será igual ao fornecido em *ObjectString*.

Se esse campo for omitido (ou seja ResObjectString.VSBufSize for zero), o *ResObjectString* não será retornado, mas o comprimento será retornado em ResObjectString.VSLength.

Se o comprimento do buffer (fornecido em ResObjectStrng.VSBufSize) for menor que o integral *ResObjectString*, a sequência será truncada e retornará quantos caracteres mais à direita couber no buffer fornecido.

Se *ResObjectString* for especificado incorretamente, de acordo com a descrição de como usar a estrutura MQCHARV ou se exceder o comprimento máximo, a chamada falhará com o código de razão MQRC\_RES\_OBJECT\_STRING\_ERROR.

### **ResolvedType (MQLONG) para MQOD**

O tipo de objeto (base) resolvido que está sendo aberto.

Os valores possíveis são:

#### **MQOT\_Q**

O objeto resolvido é uma fila Esse valor se aplica quando uma fila é aberta diretamente ou quando uma fila de alias apontando para uma fila é aberta.

#### **MQOT\_TOPIC**

O objeto resolvido é um tópico.. Esse valor se aplica quando um tópico é aberto diretamente ou quando uma fila de alias que aponta para um objeto do tópico é aberta

#### **MQOT\_NONE**

O tipo resolvido não é uma fila, nem um tópico.

### **MQOR-Registro do objeto..**

Use a estrutura MQOR para especificar o nome da fila e o nome do gerenciador de fila de uma única fila de destino MQOR é uma estrutura de entrada para as chamadas MQOPEN e MQPUT1 .

### **Disponibilidade**

A estrutura MQOR está disponível nas plataformas a seguir:

-  AIX
-  IBM i
-  Linux
-  Windows

e para o IBM MQ MQI clients conectado a esses sistemas

### **Conjunto de caracteres e codificação**

Os dados em MQOR devem estar no conjunto de caracteres fornecido pelo atributo do gerenciador de filas do **CodedCharSetId** e pela codificação do gerenciador de fila local fornecido por MQENC\_NATIVE No entanto, se o aplicativo estiver em execução como um cliente MQI MQ , a estrutura deverá estar no conjunto de caracteres e na codificação do cliente.

## Uso

Ao fornecer uma matriz dessas estruturas na chamada MQOPEN, é possível abrir uma lista de filas; essa lista é chamada de lista de distribuição. Cada mensagem colocada usando a manipulação de filas retornada por essa chamada MQOPEN é colocada em cada uma das filas na lista, desde que a fila tenha sido aberta com êxito.

## Campos

**Nota:** Na tabela a seguir, os campos são agrupados por uso e não alfabeticamente. Os tópicos filhos seguem a mesma sequência.

Tabela 505. Campos em MQOR por MQOR		
Nome e descrição do campo	Nome da constante	Valor inicial (se houver) da constante
<u>ObjectName</u> (nome do objeto)	Nenhum	Cadeia nula ou espaços em branco
<u>ObjectQMgrName</u> (nome do gerenciador de fila de objetos)	Nenhum	Cadeia nula ou espaços em branco

**Notas:**

1. O valor Sequência nula ou espaços em branco indica a sequência nula em C e caracteres em branco em outras linguagens de programação.
2. Na linguagem de programação C, a variável macroMQOR\_DEFAULT contém os valores listados na tabela. Ele pode ser usado da seguinte maneira para fornecer valores iniciais para os campos na estrutura:

```
MQOR MyOR = {MQOR_DEFAULT};
```

## Declarações de idiomas

Declaração C para MQOR

```
typedef struct tagMQOR MQOR;
struct tagMQOR {
    MQCHAR48 ObjectName; /* Object name */
    MQCHAR48 ObjectQMgrName; /* Object queue manager name */
};
```

Declaração COBOL para MQOR

```
** MQOR structure
10 MQOR.
** Object name
15 MQOR-OBJECTNAME PIC X(48).
** Object queue manager name
15 MQOR-OBJECTQMGRNAME PIC X(48).
```

Declaração PL/I para MQOR

```
dcl
1 MQOR based,
3 ObjectName char(48), /* Object name */
3 ObjectQMgrName char(48); /* Object queue manager name */
```

## Declaração Visual Basic para MQOR

```
Type MQOR
  ObjectName      As String*48 'Object name'
  ObjectQMgrName As String*48 'Object queue manager name'
End Type
```

### **ObjectName (MQCHAR48) para MQOR**

É o mesmo que o campo *ObjectName* na estrutura MQOD (consulte MQOD para obter detalhes), exceto que:

- Ele deve ser o nome de uma fila
- Não deve ser o nome de uma fila modelo.

Isto é sempre um campo de entrada. O valor inicial deste campo é a cadeia nula em C e 48 caracteres em branco em outras linguagens de programação.

### **ObjectQMgrNome (MQCHAR48) para MQOR**

Este é o mesmo que o campo *ObjectQMgrName* na estrutura MQOD (consulte MQOD para obter detalhes)

Isto é sempre um campo de entrada. O valor inicial deste campo é a cadeia nula em C e 48 caracteres em branco em outras linguagens de programação.

## **MQPD-Descrição de propriedades**

A estrutura **MQPD** é usada para definir os atributos de uma propriedade. A estrutura é um parâmetro de entrada / saída na chamada MQSETMP e um parâmetro de saída na chamada MQINQMP.

### **Disponibilidade**

A estrutura **MQPD** está disponível nas plataformas a seguir:

-  AIX
-  IBM i
-  Linux
-  Windows
-  z/OS

e para IBM MQ MQI clients..

### **Conjunto de caracteres e codificação**

Os dados em **MQPD** devem estar no conjunto de caracteres do aplicativo e codificação do aplicativo (**MQENC\_NATIVE**).

### **Campos**

**Nota:** Na tabela a seguir, os campos são agrupados por uso e não alfabeticamente. Os tópicos filhos seguem a mesma sequência.

Tabela 506. Campos em MQPD

Nome e descrição do campo	Nome da constante	Valor inicial (se houver) da constante
StrucId (identificador de estrutura)	MQPD_STRUC_ID	'PD'
Versão (número da versão da estrutura).	MQPD_VERSION_1	1
Opções (opções).	MQPD_NONE	0
Suporte (suporte necessário para propriedade de mensagem)	MQPD_SUPPORT_OPTIONAL	0
Contexto (contexto da mensagem ao qual a propriedade pertence)	MQPD_NO_CONTEXT	0
CopyOptions (opções de cópia às quais a propriedade pertence)	MQCOPY_DEFAULT	0
<p><b>Notas:</b></p> <p>1. Na linguagem de programação C, a variável de macro MQPD_DEFAULT contém os valores listados na tabela. Ele pode ser usado da seguinte maneira para fornecer valores iniciais para os campos na estrutura:</p> <pre>MQPD MyPD = {MQPD_DEFAULT};</pre>		

## Declarações de idiomas

### Declaração C para MQPD

```
typedef struct tagMQPD MQPD;
struct tagMQPD {
    MQCHAR4  StrucId;      /* Structure identifier */
    MQLONG   Version;     /* Structure version number */
    MQLONG   Options;     /* Options that control the action of
                          MQSETMP and MQINQMP */
    MQLONG   Support;     /* Property support option */
    MQLONG   Context;    /* Property context */
    MQLONG   CopyOptions; /* Property copy options */
};
```

### Declaração COBOL para MQPD

```
** MQPD structure
10 MQPD.
** Structure identifier
15 MQPD-STRUCID PIC X(4).
** Structure version number
15 MQPD-VERSION PIC S9(9) BINARY.
** Options that control the action of MQSETMP and
** MQINQMP
15 MQPD-OPTIONS PIC S9(9) BINARY.
** Property support option
15 MQPD-SUPPORT PIC S9(9) BINARY.
** Property context
15 MQPD-CONTEXT PIC S9(9) BINARY.
** Property copy options
15 MQPD-COPYOPTIONS PIC S9(9) BINARY.
```

### Declaração PL/I para MQPD

```
dcl
1 MQPD based,
3 StrucId char(4), /* Structure identifier */
```

```

3 Version      fixed bin(31), /* Structure version number */
3 Options      fixed bin(31), /* Options that control the action
                        of MQSETMP and MQINQMP */
3 Support      fixed bin(31), /* Property support option */
3 Context      fixed bin(31), /* Property context */
3 CopyOptions  fixed bin(31); /* Property copy options */

```

## Declaração do High Level Assembler para MQPD

```

MQPD          DSECT
MQPD_STRUCID  DS    CL4    Structure identifier
MQPD_VERSION  DS    F      Structure version number
MQPD_OPTIONS  DS    F      Options that control the
*              action of MQSETMP and MQINQMP
MQPD_SUPPORT  DS    F      Property support option
MQPD_CONTEXT  DS    F      Property context
MQPD_COPYOPTIONS DS    F    Property copy options
MQPD_LENGTH  EQU    *-MQPD
MQPD_AREA     DS    CL(MQPD_LENGTH)

```

### **StrucId (MQCHAR4) para MQPD**

Este é o identificador da estrutura do descritor de propriedade. Ele é sempre um campo de entrada Seu valor é MQPD\_STRUC\_ID.

O valor deve ser:.

#### **MQPD\_STRUC\_ID**

Identificador da estrutura do descritor de propriedade.

Para a linguagem de programação C, a constante **MQPD\_STRUC\_ID\_ARRAY** também é definida.. Isso tem o mesmo valor que **MQPD\_STRUC\_ID**, mas é uma matriz de caracteres em vez de uma sequência..

### **Versão (MQLONG) para MQPD**

Este é o número da versão da estrutura; o valor deve ser:

#### **MQPD\_VERSION\_1**

Estrutura do descritor da propriedade versão 1.

A constante a seguir especifica o número da versão atual:

#### **MQPD\_CURRENT\_VERSION**

Versão atual da estrutura do descritor da propriedade.

Isto é sempre um campo de entrada. O valor inicial desse campo é **MQPD\_VERSION\_1**.

### **Opções (MQLONG) para MQPD**

O valor deve ser:.

#### **MQPD\_NONE**

Nenhuma opção especificada

Isto é sempre um campo de entrada. O valor inicial desse campo é MQPD\_NONE.

### **Suporte (MQLONG) para MQPD**

Este campo descreve qual nível de suporte para a propriedade de mensagens é necessário para o gerenciador de filas, para que a mensagem que contém esta propriedade seja colocada em uma fila Isso se aplica apenas às propriedades definidas pelo IBM MQ; o suporte para todas as outras propriedades é opcional

O campo é configurado automaticamente para o valor correto quando a propriedade definida por IBM MQ é conhecida pelo gerenciador de filas. Se a propriedade não for reconhecida, MQPD\_SUPPORT\_OPTIONAL será designado.. Quando um gerenciador de filas recebe uma mensagem

contendo uma propriedade definida pelo IBM MQ que o gerenciador de filas reconhece como incorreta, o gerenciador de filas corrige o valor do campo *Support*.

Ao configurar uma propriedade definida pelo IBM MQ usando a chamada MQSETMP em uma manipulação de mensagem em que a opção MQCMHO\_NO\_VALIDATION foi configurada, o *Support* se torna um campo de entrada. Isso permite que um aplicativo coloque uma propriedade definida pelo IBM MQ, com o valor correto, em que a propriedade não é suportada pelo gerenciador de filas conectado, mas em que a mensagem deve ser processada em outro gerenciador de filas.

O valor MQPD\_SUPPORT\_OPTIONAL é sempre designado a propriedades que não são propriedades definidas pelo IBM MQ.

Se um gerenciador de filas do IBM WebSphere MQ 7.0, que suporta propriedades de mensagens, receber uma propriedade que contém um valor *Support* não reconhecido, a propriedade será tratada como se:

- MQPD\_SUPPORT\_REQUIRED foi especificado se qualquer um dos valores não reconhecidos estiver contido na MQPD\_REJECT\_UNSUP\_MASK
- MQPD\_SUPPORT\_REQUIRED\_IF\_LOCAL foi especificado se algum dos valores não reconhecidos estiver contido no MQPD\_ACCEPT\_UNSUP\_IF\_XMIT\_MASK
- MQPD\_SUPPORT\_OPTIONAL foi especificado de outra forma

Um dos valores a seguir é retornado pela chamada MQINQMP ou um dos valores pode ser especificado ao usar a chamada MQSETMP em um identificador de mensagem no qual a opção MQCMHO\_NO\_VALIDATION está configurada:

#### **MQPD\_SUPPORT\_OPTIONAL**

A propriedade é aceita por um gerenciador de filas mesmo se ela não for suportada. A propriedade pode ser descartada para a mensagem fluir para um gerenciador de filas que não suporta propriedades de mensagens. Esse valor também é designado às propriedades que não são definidas por IBM MQ.

#### **MQPD\_SUPPORT\_REQUIRED**

É necessário suporte para a propriedade. A mensagem é rejeitada por um gerenciador de filas que não suporta a propriedade definida pelo IBM MQ. A chamada MQPUT ou MQPUT1 falha com o código de conclusão MQCC\_FAILED e o código de razão MQRC\_UNSUPPORTED\_PROPERTY.

#### **MQPD\_SUPPORT\_REQUIRED\_IF\_LOCAL**

A mensagem será rejeitada por um gerenciador de filas que não suporta a propriedade definida por IBM MQ se a mensagem for destinada para uma fila local. A chamada MQPUT ou MQPUT1 falha com o código de conclusão MQCC\_FAILED e o código de razão MQRC\_UNSUPPORTED\_PROPERTY.

A chamada MQPUT ou MQPUT1 é bem-sucedida se a mensagem é destinada para um gerenciador de filas remotas.

Este é um campo de saída na chamada MQINQMP e um campo de entrada na chamada MQSETMP se a manipulação de mensagem foi criada com a opção MQCMHO\_NO\_VALIDATION configurada. O valor inicial desse campo é MQPD\_SUPPORT\_OPTIONAL.

### **Contexto (MQLONG) para MQPD**

Isto descreve a qual contexto de mensagem a propriedade pertence.

Quando um gerenciador de filas recebe uma mensagem contendo uma propriedade definida pelo IBM MQ que o gerenciador de filas reconhece como incorreta, o gerenciador de filas corrige o valor do campo *Context*.

A seguinte opção pode ser especificada:

#### **MQPD\_USER\_CONTEXT**

A propriedade é associada com o contexto do usuário.

Não é necessária nenhuma autorização especial para poder definir uma propriedade associada ao contexto do usuário utilizando a chamada MQSETMP.



Em um gerenciador de filas IBM WebSphere MQ 7.0, uma propriedade associada ao contexto do usuário é salva conforme descrito para MQOO\_SAVE\_ALL\_CONTEXT. Uma chamada MQPUT com MQPMO\_PASS\_ALL\_CONTEXT especificado, faz com que a propriedade seja copiada do contexto salvo na nova mensagem.

Se a opção descrita anteriormente não for necessária, a seguinte opção poderá ser usada:

#### **MQPD\_NO\_CONTEXT**

A propriedade não é associada com um contexto de mensagem.

Um valor não reconhecido é rejeitado com um código *Reason* de MQRC\_PD\_ERROR

Este é um campo de entrada / saída para a chamada MQSETMP e um campo de saída da chamada MQINQMP. O valor inicial desse campo é MQPD\_NO\_CONTEXT.

#### **CopyOptions (MQLONG) para MQPD**

Isso descreve em qual tipo de mensagens a propriedade deve ser copiada. Este é um campo de saída somente para propriedades IBM MQ definidas reconhecidas; IBM MQ configura o valor apropriado.

Quando um gerenciador de filas recebe uma mensagem contendo uma propriedade definida pelo IBM MQ que o gerenciador de filas reconhece como incorreta, o gerenciador de filas corrige o valor do campo *CopyOptions*.

É possível especificar uma ou mais dessas opções Para especificar mais de uma opção, inclua os valores juntos (não inclua a mesma constante mais de uma vez) ou combine os valores usando a operação OR bit a bit (se a linguagem de programação suportar operações de bit).

#### **MQCOPY\_FORWARD**

Esta propriedade é copiada para uma mensagem que está sendo encaminhada

#### **MQCOPY\_PUBLISH**

Esta propriedade é copiada na mensagem recebida por um assinante quando uma mensagem está sendo publicada.

#### **MQCOPY\_REPLY**

Esta propriedade é copiada em uma mensagem de resposta.

#### **MQCOPY\_REPORT**

Esta propriedade é copiada em uma mensagem de relatório.

#### **MQCOPY\_ALL**

Esta propriedade é copiada em todos os tipos de mensagens subsequentes.

**Opção padrão:** A seguinte opção pode ser especificada para fornecer o conjunto padrão de opções de cópia:

#### **MQCOPY\_DEFAULT**

Essa propriedade é copiada em uma mensagem sendo encaminhada, em uma mensagem de relatório ou em uma mensagem recebida por um assinante quando uma mensagem está sendo publicada.

Isso é equivalente a especificar a combinação de opções MQCOPY\_FORWARD, mais MQCOPY\_REPORT, mais MQCOPY\_PUBLISH.

Se nenhuma das opções descritas anteriormente for necessária, use a opção a seguir:

#### **MQCOPY\_NONE**

Use esse valor para indicar que nenhuma outra opção de cópia foi especificada; programaticamente não existe nenhum relacionamento entre essa propriedade e mensagens subsequentes. Isto é sempre retornado para propriedades do descritor de mensagens.

Este é um campo de entrada / saída para a chamada MQSETMP e um campo de saída da chamada MQINQMP. O valor inicial desse campo é MQCOPY\_DEFAULT.

## MQPMO-Opções de mensagem de colocação

A estrutura MQPMO permite que o aplicativo especifique opções que controlem como as mensagens são colocadas em filas ou publicadas em tópicos A estrutura é um parâmetro de entrada / saída nas chamadas MQPUT e MQPUT1 .

### Versão

A versão atual de MQPMO é MQPMO\_VERSION\_3.. Determinados campos estão disponíveis apenas em determinadas versões de MQPMO Se você precisar portar aplicativos entre vários ambientes, deverá assegurar que a versão do MQPMO seja consistente em todos os ambientes. Os campos que existem apenas em versões específicas da estrutura são identificados como tal neste tópico e nas descrições de campo

Os arquivos de cabeçalho, COPY e INCLUDE fornecidos para as linguagens de programação suportadas contêm a versão mais recente do MQPMO que é suportado pelo ambiente, mas com o valor inicial do campo *Version* configurado como MQPMO\_VERSION\_1. Para usar campos que não estão presentes na estrutura version-1 , o aplicativo deve configurar o campo *Version* para o número da versão necessária.

### Conjunto de caracteres e codificação

Os dados em MQPMO devem estar no conjunto de caracteres fornecido pelo atributo do gerenciador de fila do **CodedCharSetId** e pela codificação do gerenciador de fila local fornecido por MQENC\_NATIVE No entanto, se o aplicativo estiver em execução como um cliente MQI MQ , a estrutura deverá estar no conjunto de caracteres e na codificação do cliente.

### Campos

**Nota:** Na tabela a seguir, os campos são agrupados por uso e não alfabeticamente. Os tópicos filhos seguem a mesma sequência.

Nome e descrição do campo	Nome da constante	Valor inicial (se houver) da constante
<u>StrucId</u> (identificador de estrutura)	MQPMO_STRUC_ID	'PMO→'
<u>Versão</u> (número da versão da estrutura).	MQPMO_VERSION_1	1
<u>Opções</u> (opções que controlam a ação de MQPUT e MQPUT1)	MQPMO_NONE	0
<u>Tempo Limite</u> (reservado)	Nenhum	-1
<u>Contexto</u> (manipulação de objetos da fila de entrada)..	Nenhum	0
<u>KnownDestCount</u> (número de mensagens enviadas com sucesso para filas locais)	Nenhum	0
<u>UnknownDestCount</u> (número de mensagens enviadas com sucesso para filas remotas)	Nenhum	0
<u>InvalidDestCount</u> (número de mensagens que não puderam ser enviadas)	Nenhum	0
<u>ResolvedQName</u> (nome resolvido da fila de destino).	Nenhum	Cadeia nula ou espaços em branco
<u>ResolvedQMgrNome</u> (nome resolvido do gerenciador de filas de destino)	Nenhum	Cadeia nula ou espaços em branco

Tabela 507. Campos em MQPMO (continuação)		
Nome e descrição do campo	Nome da constante	Valor inicial (se houver) da constante
<b>Nota:</b> Os campos restantes serão ignorados se <i>Version</i> for menor que MQPMO_VERSION_2.		
RecsPresent (número de registros de mensagens put ou registros de resposta presentes)	Nenhum	0
PutMsgRecFields (sinalizadores indicando quais campos MQPMPR estão presentes)	MQPMRF_NONE	0
PutMsgRecOffset (deslocamento do primeiro registro de mensagem put do início de MQPMO)	Nenhum	0
ResponseRecDeslocamento (deslocamento do primeiro registro de resposta do início de MQPMO)	Nenhum	0
PutMsgRecPtr (endereço do primeiro registro de mensagem put)	Nenhum	Ponteiro nulo ou bytes nulos
ResponseRecPtr (endereço do primeiro registro de resposta);	Nenhum	Ponteiro nulo ou bytes nulos
<b>Nota:</b> Os campos restantes serão ignorados se <i>Version</i> for menor que MQPMO_VERSION_3..		
OriginalMsgHandle (identificador de mensagem original)	MQHM_NONE	0
NewMsgHandle (nova manipulação de mensagens)	MQHM_NONE	0
Ação (tipo de put que está sendo executado e o relacionamento entre a mensagem original especificada pelo campo <i>OriginalMsgHandle</i> e a nova mensagem especificada pelo campo <i>NewMsgHandle</i> )	MQACTP_NEW	0
PubLevel (nível de assinatura destinado pela publicação)	Nenhum	9
<b>Notas:</b> <ol style="list-style-type: none"> <li>1. O símbolo ~ representa um único caractere em branco.</li> <li>2. O valor Sequência nula ou espaços em branco indica a sequência nula em C e caracteres em branco em outras linguagens de programação.</li> <li>3. Na linguagem de programação C, a variável macro MQPMO_DEFAULT contém os valores listados na tabela. Use-o da seguinte maneira para fornecer valores iniciais para os campos na estrutura: <pre>MQPMO MyPMO = {MQPMO_DEFAULT};</pre> </li> </ol>		

## Declarações de idiomas

### Declaração C para MQPMO

```
typedef struct tagMQPMO MQPMO;
struct tagMQPMO {
    MQCHAR4   StrucId;           /* Structure identifier */
    MQLONG    Version;          /* Structure version number */
    MQLONG    Options;          /* Options that control the action of
                                MQPUT and MQPUT1 */
    MQLONG    Timeout;          /* Reserved */
};
```

```

MQHOBJ      Context;          /* Object handle of input queue */
MQLONG      KnownDestCount;  /* Number of messages sent
                               successfully to local queues */
MQLONG      UnknownDestCount; /* Number of messages sent
                               successfully to remote queues */
MQLONG      InvalidDestCount; /* Number of messages that could not
                               be sent */
MQCHAR48    ResolvedQName;   /* Resolved name of destination
                               queue */
MQCHAR48    ResolvedQMgrName; /* Resolved name of destination queue
                               manager */
/* Ver:1 */
MQLONG      RecsPresent;     /* Number of put message records or
                               response records present */
MQLONG      PutMsgRecFields; /* Flags indicating which MQPMR fields
                               are present */
MQLONG      PutMsgRecOffset; /* Offset of first put message record
                               from start of MQPMO */
MQLONG      ResponseRecOffset; /* Offset of first response record
                               from start of MQPMO */
MQPTR       PutMsgRecPtr;    /* Address of first put message
                               record */
MQPTR       ResponseRecPtr;  /* Address of first response record */
/* Ver:2 */
MQHMSG      OriginalMsgHandle; /* Original message handle */
MQHMSG      NewMsgHandle;     /* New message handle */
MQLONG      Action;          /* The action being performed */
MQLONG      PubLevel;        /* Subscription level */
/* Ver:3 */
};

```

## Declaração COBOL para MQPMO

```

** MQPMO structure
10 MQPMO.
** Structure identifier
15 MQPMO-STRUCID          PIC X(4).
** Structure version number
15 MQPMO-VERSION        PIC S9(9) BINARY.
** Options that control the action of MQPUT and MQPUT1
15 MQPMO-OPTIONS        PIC S9(9) BINARY.
** Reserved
15 MQPMO-TIMEOUT        PIC S9(9) BINARY.
** Object handle of input queue
15 MQPMO-CONTEXT        PIC S9(9) BINARY.
** Number of messages sent successfully to local queues
15 MQPMO-KNOWNDSTCOUNT PIC S9(9) BINARY.
** Number of messages sent successfully to remote queues
15 MQPMO-UNKNOWNDSTCOUNT PIC S9(9) BINARY.
** Number of messages that could not be sent
15 MQPMO-INVALIDDESTCOUNT PIC S9(9) BINARY.
** Resolved name of destination queue
15 MQPMO-RESOLVEDQNAME  PIC X(48).
** Resolved name of destination queue manager
15 MQPMO-RESOLVEDQMGRNAME PIC X(48).
** Number of put message records or response records present
15 MQPMO-RECSPRESENT   PIC S9(9) BINARY.
** Flags indicating which MQPMR fields are present
15 MQPMO-PUTMSGRECFIELDS PIC S9(9) BINARY.
** Offset of first put message record from start of MQPMO
15 MQPMO-PUTMSGRECOFFSET PIC S9(9) BINARY.
** Offset of first response record from start of MQPMO
15 MQPMO-RESPONSERECOFFSET PIC S9(9) BINARY.
** Address of first put message record
15 MQPMO-PUTMSGRECPTTR  POINTER.
** Address of first response record
15 MQPMO-RESPONSERECPTTR POINTER.
** Original message handle
15 MQPMO-ORIGINALMSGHANDLE PIC S9(18) BINARY.
** New message handle
15 MQPMO-NEWMMSGHANDLE  PIC S9(18) BINARY.
** The action being performed
15 MQPMO-ACTION         PIC S9(9) BINARY.
** Publish level
15 MQPMO-PUBLEVEL      PIC S9(9) BINARY.

```

## Declaração PL/I para MQPMO

```

dcl
  1 MQPMO based,
  3 StrucId          char(4),          /* Structure identifier */
  3 Version          fixed bin(31),    /* Structure version number */
  3 Options          fixed bin(31),    /* Options that control the action
                                     of MQPUT and MQPUT1 */
  3 Timeout          fixed bin(31),    /* Reserved */
  3 Context          fixed bin(31),    /* Object handle of input queue */
  3 KnownDestCount  fixed bin(31),    /* Number of messages sent
                                     successfully to local queues */
  3 UnknownDestCount fixed bin(31),    /* Number of messages sent
                                     successfully to remote queues */
  3 InvalidDestCount fixed bin(31),    /* Number of messages that could
                                     not be sent */
  3 ResolvedQName   char(48),         /* Resolved name of destination
                                     queue */
  3 ResolvedQMgrName char(48),        /* Resolved name of destination
                                     queue manager */
  3 RecsPresent     fixed bin(31),    /* Number of put message records or
                                     response records present */
  3 PutMsgRecFields fixed bin(31),    /* Flags indicating which MQPMR
                                     fields are present */
  3 PutMsgRecOffset fixed bin(31),    /* Offset of first put message
                                     record from start of MQPMO */
  3 ResponseRecOffset fixed bin(31),  /* Offset of first response record
                                     from start of MQPMO */
  3 PutMsgRecPtr    pointer,          /* Address of first put message
                                     record */
  3 ResponseRecPtr  pointer,          /* Address of first response
                                     record */
  3 OriginalMsgHandle fixed bin(63),  /* Original message handle */
  3 NewMsgHandle    fixed bin(63);    /* New message handle */
  3 Action          fixed bin(31);    /* The action being performed */
  3 PubLevel       fixed bin(31);    /* Publish level */

```

## Declaração High Level Assembler para MQPMO

```

MQPMO          DSECT
MQPMO_STRUCID  DS   CL4   Structure identifier
MQPMO_VERSION  DS   F     Structure version number
MQPMO_OPTIONS  DS   F     Options that control the action of
*               MQPUT and MQPUT1
MQPMO_TIMEOUT  DS   F     Reserved
MQPMO_CONTEXT  DS   F     Object handle of input queue
MQPMO_KNOWNDESTCOUNT DS F   Number of messages sent successfully
*               to local queues
MQPMO_UNKNOWNDESTCOUNT DS F   Number of messages sent successfully
*               to remote queues
MQPMO_INVALIDDESTCOUNT DS F   Number of messages that could not be
*               sent
MQPMO_RESOLVEDQNAME DS   CL48  Resolved name of destination queue
MQPMO_RESOLVEDQMGRNAME DS   CL48  Resolved name of destination queue
*               manager
MQPMO_RECSPRESENT DS   F     Number of put message records or
*               response records present
MQPMO_PUTMSGRECFIELDS DS   F     Flags indicating which MQPMR
*               fields are present
MQPMO_PUTMSGRECOFFSET DS   F     Offset of first put message record
*               from start of MQPMO
MQPMO_RESPONSERECOFFSET DS   F     Offset of first response record
*               from start of MQPMO
MQPMO_PUTMSGRECPTTR DS   F     Address of first put message
*               record
MQPMO_RESPONSERECPTTR DS   F     Address of first response record
MQPMO_ORIGINALMSGHANDLE DS   D     Original message handle
MQPMO_NEWMSGHANDLE DS   D     New message handle
MQPMO_ACTION   DS   F     The action being performed
MQPMO_PUBLEVEL DS   F     Publish level
*
MQPMO_LENGTH   EQU   *-MQPMO
MQPMO_AREA     DS   CL(MQPMO_LENGTH)

```

## Declaração do Visual Basic para MQPMO.

```
Type MQPMO
  StructId      As String*4  'Structure identifier'
  Version       As Long     'Structure version number'
  Options       As Long     'Options that control the action of'
                  'MQPUT and MQPUT1'
  Timeout       As Long     'Reserved'
  Context       As Long     'Object handle of input queue'
  KnownDestCount As Long     'Number of messages sent successfully'
                  'to local queues'
  UnknownDestCount As Long   'Number of messages sent successfully'
                  'to remote queues'
  InvalidDestCount As Long   'Number of messages that could not be'
                  'sent'
  ResolvedQName As String*48 'Resolved name of destination queue'
  ResolvedQMgrName As String*48 'Resolved name of destination queue'
                  'manager'
  RecsPresent   As Long     'Number of put message records or'
                  'response records present'
  PutMsgRecFields As Long   'Flags indicating which MQPMR fields'
                  'are present'
  PutMsgRecOffset As Long   'Offset of first put message record'
                  'from start of MQPMO'
  ResponseRecOffset As Long 'Offset of first response record from'
                  'start of MQPMO'
  PutMsgRecPtr   As MQPTR   'Address of first put message record'
  ResponseRecPtr As MQPTR   'Address of first response record'
End Type
```

### **StructId (MQCHAR4) para MQPMO**

Este é o identificador de estrutura da estrutura de opções de mensagem de colocação Ele é sempre um campo de entrada Seu valor é MQPMO\_STRUC\_ID.

O valor deve ser:.

#### **MQPMO\_STRUC\_ID**

Identificador para a estrutura de opções de mensagem put.

Para a linguagem de programação C, a constante MQPMO\_STRUC\_ID\_ARRAY também é definida Ele tem o mesmo valor que MQPMO\_STRUC\_ID, mas é uma matriz de caracteres em vez de uma cadeia de caracteres.

### **Versão (MQLONG) para MQPMO**

Número de versão da estrutura.

O valor deve ser um dos seguintes:

#### **MQPMO\_VERSION\_1**

Version-1 estrutura de opções put-message.

Esta versão é suportada em todos os ambientes

#### **MQPMO\_VERSION\_2**

Version-2 estrutura de opções put-message.

Esta versão é suportada nos seguintes ambientes:

-  AIX
-  IBM i
-  Linux
-  Windows

e para o IBM MQ MQI clients conectado a esses sistemas

#### **MQPMO\_VERSION\_3**

Version-3 estrutura de opções put-message.

Esta versão é suportada em todos os ambientes

Os campos que existem apenas na versão mais recente da estrutura são identificados como tal nas descrições dos campos A constante a seguir especifica o número da versão atual:

### **MQPMO\_CURRENT\_VERSION**

Versão atual da estrutura de opções de put-message

Isto é sempre um campo de entrada. O valor inicial desse campo é MQPMO\_VERSION\_1.

### **Opções (MQLONG) para MQPMO**

O campo Opções controla a operação de chamadas **MQPUT** e **MQPUT1**.

**Opção de Escopo** É possível especificar alguma ou nenhuma das opções de MQPMO Para especificar mais de uma opção, inclua os valores juntos (não inclua a mesma constante mais de uma vez) ou combine os valores usando a operação OR bit a bit (se a linguagem de programação suportar operações de bit). As combinações que não são válidas são observadas; quaisquer outras combinações são válidas

A opção a seguir controla o escopo das publicações enviadas:

### **MQPMO\_SCOPE\_QMGR**

A publicação é enviada apenas para os assinantes que se inscreveram neste gerenciador de filas A publicação não é encaminhada para nenhum gerenciador de fila de publicação / assinatura remoto que tenha feito uma assinatura para esse gerenciador de fila, que substitui qualquer comportamento que tenha sido configurado usando o atributo de tópico PUBSCOPE

**Nota:** Se não for configurado, o escopo da publicação será determinado pelo atributo do tópico PUBSCOPE

**Opções de publicação..** As opções a seguir controlam a maneira como as mensagens são publicadas em um tópico:

### **MQPMO\_SUPPRESS\_REPLYTO**

Qualquer informação especificada nos campos *ReplyToQ* e *ReplyToQMGR* do MQMD desta publicação não é transmitida aos assinantes. Se essa opção for usada com uma opção de relatório que requeira um *ReplyToQ*, a chamada falhará com MQRC\_MISSING\_REPLY\_TO\_Q

### **MQPMO\_RETAIN**

A publicação que está sendo enviada deve ser retida pelo gerenciador de filas. Essa retenção permite que um assinante solicite uma cópia dessa publicação após o horário em que ela foi publicada, usando a chamada MQSUBRQ.. Também permite que uma publicação seja enviada aos aplicativos que fazem suas assinaturas após o momento que esta publicação foi feita (a menos que eles escolham não enviá-la usando a opção MQSO\_NEW\_PUBLICATIONS\_ONLY). Se um aplicativo receber uma publicação que foi retida, ela será indicada pela propriedade de mensagem MQIsRetained dessa publicação.

Somente uma publicação pode ser retida em cada nó da árvore de tópicos. Portanto, se já houver uma publicação retida para este tópico, publicada por qualquer outro aplicativo, ela será substituída por esta publicação Portanto, é melhor evitar que mais de um publicador mantenha mensagens no mesmo tópico.

Quando as publicações retidas são solicitadas por um assinante, a assinatura usada pode conter um curinga no tópico, nesse caso, várias publicações retidas podem corresponder (em vários nós na árvore de tópicos) e várias publicações podem ser enviadas para o aplicativo solicitante. Consulte a descrição da chamada "[MQSUBRQ-Pedido de assinatura](#)" na página 824 para obter mais detalhes..

Para obter informações sobre como as publicações retidas interagem com níveis de assinatura, consulte [Interceptando publicações](#).

Se essa opção for usada e a publicação não puder ser retida, a mensagem não será publicada e a chamada falhará com MQRC\_PUT\_NOT\_RETAINED.

### **MQPMO\_NOT\_OWN\_SUBS**

Informa ao gerenciador de filas que o aplicativo não deseja enviar nenhuma de suas publicações para as assinaturas que ele possui. As assinaturas são consideradas de propriedade do mesmo aplicativo se as identificações de conexão forem as mesmas.

### **MQPMO\_WARN\_IF\_NO\_SUBS\_MATCHED**

Se nenhuma assinatura corresponder à publicação, retorne um código de conclusão (*CompCode*) de MQCC\_WARNING e o código de razão MQRC\_NO\_SUBS\_MATCHED.

Se MQRC\_NO\_SUBS\_MATCHED for retornado pela operação de colocação, a publicação não foi entregue a nenhuma assinatura. No entanto, se a opção MQPMO\_RETAIN for especificada na operação put, a mensagem será retida e entregue a qualquer subscrição correspondente definida subsequentemente.

Uma assinatura no tópico corresponde à publicação se qualquer uma das seguintes condições for atendida:

- A mensagem é entregue à fila de assinaturas.
- A mensagem teria sido entregue à fila de assinaturas, mas um problema com a fila significa que a mensagem não pode ser colocada na fila e, conseqüentemente, foi colocada na fila de devoluções ou descartada.
- É definida uma saída de roteamento que suprime a entrega da mensagem para a assinatura.

Uma assinatura no tópico não corresponde à publicação se qualquer uma das condições a seguir for atendida:

- A assinatura possui uma seqüência de seleção que não corresponde à publicação.
- A assinatura especificou a opção MQSO\_PUBLICATION\_ON\_REQUEST.
- A publicação não foi entregue porque a opção MQPMO\_NOT\_OWN\_SUBS foi especificada na operação put e a assinatura corresponde à identidade do publicador.

**Opções de Ponto de Sincronização** As opções a seguir estão relacionadas à participação da chamada MQPUT ou MQPUT1 em uma unidade de trabalho:

### **MQPMO\_SYNCPOINT**

A solicitação é operada dentro dos protocolos normais de unidade de trabalho. A mensagem não é visível fora da unidade de trabalho até que a unidade de trabalho seja confirmada. Se a unidade de trabalho for recuperada, a mensagem será excluída.

Se MQPMO\_SYNCPOINT e MQPMO\_NO\_SYNCPOINT não forem especificados, a inclusão do pedido put em protocolos de unidade de trabalho será determinada pelo ambiente que executa o gerenciador de filas e não pelo ambiente que executa o aplicativo. No z/OS, a solicitação de colocação está dentro de uma unidade de trabalho. Em todos os outros ambientes, a solicitação put não está em uma unidade de trabalho.

Devido a essas diferenças, um aplicativo que você deseja transportar não deve permitir que essa opção seja padrão; especifique MQPMO\_SYNCPOINT ou MQPMO\_NO\_SYNCPOINT explicitamente.

Não especifique MQPMO\_SYNCPOINT com MQPMO\_NO\_SYNCPOINT.

### **MQPMO\_NO\_SYNCPOINT**

O pedido é operado fora dos protocolos normais de unidade de trabalho. A mensagem está disponível imediatamente e não pode ser excluída efetuando backout de uma unidade de trabalho.

Se MQPMO\_NO\_SYNCPOINT e MQPMO\_SYNCPOINT não forem especificados, a inclusão da solicitação put em protocolos de unidade de trabalho será determinada pelo ambiente executando o gerenciador de filas e não pelo ambiente executando o aplicativo. No z/OS, a solicitação de colocação está dentro de uma unidade de trabalho. Em todos os outros ambientes, a solicitação put não está em uma unidade de trabalho.

Devido a essas diferenças, um aplicativo que você deseja transportar não deve permitir que essa opção seja padrão; especifique MQPMO\_SYNCPOINT ou MQPMO\_NO\_SYNCPOINT explicitamente.

Não especifique MQPMO\_NO\_SYNCPOINT com MQPMO\_SYNCPOINT..



**Opções de identificador de mensagens e identificador de correlação.** As opções a seguir solicitam que o gerenciador de filas gere um novo identificador de mensagens ou identificador de correlação:

#### **MQPMO\_NEW\_MSG\_ID**

O gerenciador de filas substitui o conteúdo do campo *MsgId* no MQMD por um novo identificador de mensagens.. Esse identificador de mensagem é enviado com a mensagem e retornado ao aplicativo na saída da chamada MQPUT ou MQPUT1 .

A opção MQPMO\_NEW\_MSG\_ID também pode ser especificada quando a mensagem está sendo colocada em uma lista de distribuições; consulte a descrição do campo *MsgId* na estrutura MQPMR para obter detalhes.

Usar essa opção alivia a necessidade de reconfigurar o campo *MsgId* para MQMI\_NONE antes de cada chamada MQPUT ou MQPUT1 .

#### **MQPMO\_NEW\_CORREL\_ID**

O gerenciador de filas substitui o conteúdo do campo *CorrelId* no MQMD por um novo identificador de correlação. Esse identificador de correlação é enviado com a mensagem e retornado ao aplicativo na saída da chamada MQPUT ou MQPUT1 .

A opção MQPMO\_NEW\_CORREL\_ID também pode ser especificada quando a mensagem está sendo colocada em uma lista de distribuição; consulte a descrição do campo *CorrelId* na estrutura MQPMR para obter detalhes

MQPMO\_NEW\_CORREL\_ID é útil em situações em que o aplicativo requer um identificador de correlação exclusivo

**Opções de grupo e segmento.** As opções a seguir estão relacionadas ao processamento de mensagens em grupos e segmentos de mensagens lógicas. Leia as definições a seguir para ajudar a entender a opção.



**Atenção:** Você não pode usar segmentos ou mensagens agrupadas com Publicação/ Assinatura.

#### **Mensagem física**

É a menor unidade de informações que pode ser colocada ou removida de uma fila; ela geralmente corresponde às informações especificadas ou recuperadas em uma única chamada MQPUT, MQPUT1 ou MQGET. Cada mensagem física possui seu próprio descritor de mensagens (MQMD). Geralmente, as mensagens físicas são distinguidas por valores diferentes para o identificador de mensagens (campo *MsgId* em MQMD), embora isso não seja aplicado pelo gerenciador de filas.

#### **Mensagem lógica**

Uma mensagem lógica é uma única unidade de informações do aplicativo somente para plataformas não z/OS Na ausência de restrições do sistema, uma mensagem lógica é a mesma que uma mensagem física Mas onde as mensagens lógicas são extremamente grandes, as restrições do sistema podem tornar aconselhável ou necessário dividir uma mensagem lógica em duas ou mais mensagens físicas, chamadas *segmentos*.

Uma mensagem lógica que foi segmentada consiste em duas ou mais mensagens físicas que possuem o mesmo identificador de grupo não nulo (campo *GroupId* no MQMD) e o mesmo número de sequência da mensagem (campo *MsgSeqNumber* no MQMD). Os segmentos são diferenciados por valores diferentes para o deslocamento de segmento (campo *Offset* em MQMD), que fornece o deslocamento dos dados na mensagem física do início dos dados na mensagem lógica. Como cada segmento é uma mensagem física, os segmentos em uma mensagem lógica geralmente possuem identificadores de mensagens diferentes..

Uma mensagem lógica que não foi segmentada, mas para a qual a segmentação foi permitida pelo aplicativo de envio, também possui um identificador de grupo não nulo, embora nesse caso haja apenas uma mensagem física com esse identificador de grupo se a mensagem lógica não pertencer a um grupo de mensagens. Mensagens lógicas para as quais a segmentação foi inibida pelo aplicativo de envio possuem um identificador de grupo nulo (MQGI\_NONE), a menos que a mensagem lógica pertença a um grupo de mensagens.

## Grupo de mensagens

Um grupo de mensagens é um conjunto de uma ou mais mensagens lógicas que possuem o mesmo identificador de grupo não nulo. As mensagens lógicas no grupo são diferenciadas por valores diferentes para o número de seqüência da mensagem, que é um número inteiro no intervalo de 1 a  $n$ , em que  $n$  é o número de mensagens lógicas no grupo. Se uma ou mais mensagens lógicas forem segmentadas, haverá mais de  $n$  mensagens físicas no grupo.

## MQPMO\_LOGICAL\_ORDER

Esta opção informa ao gerenciador de filas como o aplicativo coloca mensagens em grupos e segmentos de mensagens lógicas... Ela só pode ser especificada na chamada MQPUT. Ela não é válida na chamada MQPUT1.

Se MQPMO\_LOGICAL\_ORDER é especificado, ele indica que o aplicativo utiliza chamadas MQPUT sucessivas para:

1. Colocar os segmentos em cada mensagem lógica na ordem crescente de deslocamento de segmento, iniciando a partir de 0, sem lacunas.
2. Colocar todos os segmentos em uma mensagem lógica antes de colocar os segmentos na próxima mensagem lógica.
3. Colocar as mensagens lógicas em cada grupo de mensagens na ordem crescente de número de seqüência da mensagem, iniciando a partir de 1, sem lacunas. IBM MQ incrementa o número de seqüências da mensagem automaticamente.
4. Colocar todas as mensagens lógicas em um grupo de mensagens antes de colocar mensagens lógicas no próximo grupo de mensagens.

Para obter informações detalhadas sobre MQPMO\_LOGICAL\_ORDER, consulte [ordenação lógica e física](#)

**Opções de Contexto** As seguintes opções controlam o processamento do contexto da mensagem:

## MQPMO\_NO\_CONTEXT

O contexto de identidade e de origem é configurado para não indicar nenhum contexto Isso significa que os campos de contexto no MQMD são configurados para:

- Espaços em branco para campos de caracteres
- Nulos para campos de bytes
- Zeros para campos numéricos

## MQPMO\_DEFAULT\_CONTEXT

A mensagem deve ter informações de contexto padrão associadas a ela, para identidade e origem. O gerenciador de filas configura os campos de contexto no descritor de mensagem da seguinte forma:

Tabela 508. Valores de informações de contexto padrão para campos MQMD

<b>Campo no MQMD</b>	<b>Valor Usado</b>
<i>UserIdentifier</i>	Determinado a partir do ambiente, se possível; caso contrário, configure como espaços em branco
<i>AccountingToken</i>	Determinado a partir do ambiente, se possível; caso contrário, configure como MQACT_NONE.
<i>AppIdentityData</i>	Configure como em branco.
<i>PutApplType</i>	Determinado a partir do ambiente.
<i>PutApplName</i>	Determinado a partir do ambiente, se possível; caso contrário, configure como espaços em branco
<i>PutDate</i>	Configure para a data em que a mensagem é colocada
<i>PutTime</i>	Configure para o horário em que a mensagem é colocada
<i>AppOriginData</i>	Configure como em branco.

Para obter mais informações sobre contexto da mensagem, consulte [Contexto da mensagem](#).

Esses são os valores e ações padrão se nenhuma opção de contexto for especificada

### **MQPMO\_PASS\_IDENTITY\_CONTEXT**

A mensagem deve ter as informações de contexto associadas a ela O contexto de identidade é obtido do identificador de filas especificado no campo *Context* . As informações de contexto de origem são geradas pelo gerenciador de filas da mesma maneira que para MQPMO\_DEFAULT\_CONTEXT (consulte a tabela anterior para obter valores). Para obter mais informações sobre contexto da mensagem, consulte [Contexto da mensagem](#).

Para a chamada MQPUT, a fila deve ter sido aberta com a opção MQOO\_PASS\_IDENTITY\_CONTEXT (ou uma opção que implica). Para a chamada MQPUT1 , a mesma verificação de autorização é executada para a chamada MQOPEN com a opção MQOO\_PASS\_IDENTITY\_CONTEXT.

### **MQPMO\_PASS\_ALL\_CONTEXT**

A mensagem deve ter as informações de contexto associadas a ela O contexto é obtido do identificador de filas especificado no campo *Context* . Para obter mais informações sobre o contexto da mensagem, consulte [Controlando Informações de Contexto](#)

Para a chamada MQPUT, a fila deve ter sido aberta com a opção MQOO\_PASS\_ALL\_CONTEXT (ou uma opção que a implica). Para a chamada MQPUT1 , a mesma verificação de autorização é executada para a chamada MQOPEN com a opção MQOO\_PASS\_ALL\_CONTEXT.

### **MQPMO\_SET\_IDENTITY\_CONTEXT**

A mensagem deve ter as informações de contexto associadas a ela O aplicativo especifica o contexto de identidade na estrutura do MQMD As informações de contexto de origem são geradas pelo gerenciador de filas da mesma maneira que para MQPMO\_DEFAULT\_CONTEXT (consulte a tabela anterior para obter valores). Para obter mais informações sobre contexto da mensagem, consulte [Contexto da mensagem](#).

Para a chamada MQPUT, a fila deve ter sido aberta com a opção MQOO\_SET\_IDENTITY\_CONTEXT (ou uma opção que a implica). Para a chamada MQPUT1 , a mesma verificação de autorização é executada para a chamada MQOPEN com a opção MQOO\_SET\_IDENTITY\_CONTEXT.

### **MQPMO\_SET\_ALL\_CONTEXT**

A mensagem deve ter as informações de contexto associadas a ela O aplicativo especifica a identidade, origem e contexto do usuário na estrutura MQMD. Para obter mais informações sobre contexto da mensagem, consulte [Contexto da mensagem](#).

Para a chamada MQPUT, a fila deve ser aberta com a opção MQOO\_SET\_ALL\_CONTEXT. Para a chamada MQPUT1 , a mesma verificação de autorização é executada para a chamada MQOPEN com a opção MQOO\_SET\_ALL\_CONTEXT.

É possível especificar apenas uma das opções de contexto MQPMO\_\*\_CONTEXT Se você especificar nenhum, MQPMO\_DEFAULT\_CONTEXT será assumido..

**Opções de propriedade** A opção a seguir está relacionada às propriedades da mensagem:

### **MQPMO\_MD\_FOR\_OUTPUT\_ONLY**

O parâmetro do descritor de mensagens deve ser utilizado apenas para a saída para retornar o descritor de mensagens da mensagem que foi colocada Os campos do descritor de mensagem associados ao *NewMsgHandle*, *OriginalMsgHandle* ou ambos os campos da estrutura **MQPMO** devem ser usados para entrada.

Se um identificador de mensagem válido não for fornecido, então a chamada falhará com o código de razão **MQRC\_MD\_ERROR**

**Opções de resposta Put.** As opções a seguir controlam a resposta retornada para uma chamada MQPUT ou MQPUT1 . É possível especificar apenas uma destas opções Se MQPMO\_ASYNC\_RESPONSE e MQPMO\_SYNC\_RESPONSE não forem especificados, MQPMO\_RESPONSE\_AS\_Q\_DEF ou MQPMO\_RESPONSE\_AS\_TOPIC\_DEF é assumido.

## **MQPMO\_ASYNC\_RESPONSE**

A opção `MQPMO_ASYNC_RESPONSE` solicita que uma operação `MQPUT` ou `MQPUT1` seja concluída sem o aplicativo esperando o gerenciador de filas concluir a chamada. O uso dessa opção pode melhorar o desempenho do sistema de mensagens, particularmente para aplicativos que usam ligações do cliente. Um aplicativo pode verificar periodicamente, usando o verbo `MQSTAT`, se ocorreu um erro durante qualquer chamada assíncrona anterior...

Com essa opção, apenas os seguintes campos são garantidos para serem preenchidos no `MQMD`;

- `ApplIdentityData`
- `PutApplType`
- `PutApplName`
- `ApplOriginData`

Além disso, se um ou ambos `MQPMO_NEW_MSG_ID` ou `MQPMO_NEW_CORREL_ID` forem especificados como opções, o `MsgId` e o `CorrelId` retornado também serão concluídos. (`MQPMO_NEW_MSG_ID` pode ser especificado implicitamente especificando um campo em branco `MsgId`).

Apenas os campos especificados anteriores são preenchidos. Outras informações que normalmente seriam retornadas na estrutura `MQMD` ou `MQPMO` são indefinidas..

Ao solicitar resposta de colocação assíncrona para `MQPUT1`, o `ResolvedQName` e `ResolvedQMgrNome` retornado na estrutura `MQOD` são indefinidos.

Ao solicitar resposta de put assíncrono para `MQPUT` ou `MQPUT1`, um `CompCode` e `Reason` de `MQCC_OK` e `MQRC_NONE` não significam necessariamente que a mensagem foi colocada com sucesso em uma fila. Ao desenvolver um aplicativo `MQI` que usa a resposta put assíncrona e requer a confirmação de que as mensagens foram colocadas em uma fila, deve-se verificar os códigos `CompCode` e `Reason` das operações put e também usar `MQSTAT` para consultar informações de erro assíncronas.

Embora o sucesso ou a falha de cada chamada `MQPUT` ou `MQPUT1` individual não seja retornado imediatamente, o primeiro erro que ocorreu sob uma chamada assíncrona pode ser determinado posteriormente por meio de uma chamada para `MQSTAT`.

Se uma mensagem persistente no ponto de sincronização falhar ao ser entregue usando a resposta put assíncrona e você tentar confirmar a transação, a confirmação falhará e a transação será restaurada com um código de conclusão de `MQCC_FAILED` e uma razão de `MQRC_BACKED_OUT`. O aplicativo pode fazer uma chamada para `MQSTAT` para determinar a causa de uma falha anterior de `MQPUT` ou `MQPUT1`.

## **MQPMO\_SYNC\_RESPONSE**

Especificar esse tipo de resposta put assegura que a operação `MQPUT` ou `MQPUT1` seja sempre emitida de forma síncrona. Se a operação put for bem-sucedida, todos os campos no `MQMD` e `MQPMO` serão preenchidos.

Essa opção assegura uma resposta síncrona independentemente do valor de resposta put padrão definido na fila ou no objeto do tópico.

## **MQPMO\_RESPONSE\_AS\_Q\_DEF**

Se esse valor for especificado para uma chamada `MQPUT`, o tipo de resposta put usado será obtido do valor `DEFPRESP` especificado na fila quando ele foi aberto pela primeira vez pelo aplicativo.

- Se a fila for uma fila de clusters, e esse valor for especificado para uma chamada `MQPUT`, o tipo de resposta put usado será obtido do atributo **DEFPRESP** definido no gerenciador de filas *destination* que possui a instância específica da fila na qual a mensagem é colocada.

Quando há múltiplas instâncias da fila de clusters e elas diferem nesse atributo, o valor de uma delas é escolhido e não pode ser previsto qual será usado. É necessário, portanto, configurar esse atributo para o mesmo valor em todas as instâncias. Se este não for o caso, a mensagem de erro `AMQ9407` será emitida para os logs do gerenciador de filas. Consulte também [Como os atributos do objeto de destino são resolvidos para filas de alias, remotas e de clusters?](#)

- Se a fila não for uma fila de clusters e esse valor for especificado para uma chamada MQPUT, o tipo de resposta put usado será obtido do atributo **DEFPRESP** definido no gerenciador de filas *local*, mesmo se o gerenciador de filas de destino for remoto.

Se um aplicativo cliente estiver conectado a um gerenciador de filas em um nível anterior ao IBM WebSphere MQ 7.0, ele se comportará como se MQPMO\_SYNC\_RESPONSE fosse especificado

Se essa opção for especificada para uma chamada MQPUT1, o valor do atributo DEFPRESP não será conhecido antes que a solicitação seja enviada ao servidor. Por padrão, se a chamada MQPUT1 estiver usando MQPMO\_SYNCPOINT, ela se comporta como para MQPMO\_ASYNC\_RESPONSE e se estiver usando MQPMO\_NO\_SYNCPOINT, ela se comporta como para MQPMO\_SYNC\_RESPONSE. No entanto, é possível substituir esse comportamento padrão configurando a propriedade Put1DefaultAlwaysSync no arquivo de configuração do cliente, consulte [Sub-rotina CHANNELS do arquivo de configuração do cliente](#).

### **MQPMO\_RESPONSE\_AS\_TOPIC\_DEF**

MQPMO\_RESPONSE\_AS\_TOPIC\_DEF é um sinônimo para MQPMO\_RESPONSE\_AS\_Q\_DEF para uso com objetos de tópicos..

**Outras opções..** As opções a seguir controlam a verificação de autorização, o que acontece quando o gerenciador de filas está em quiesce e resolvendo nomes de fila e de gerenciador de filas:

### **MQPMO\_ALTERNATE\_USER\_AUTHORITY**

MQPMO\_ALTERNATE\_USER\_AUTHORITY indica que o campo *AlternateUserId* no parâmetro **ObjDesc** da chamada MQPUT1 contém um identificador de usuário que deve ser usado para validar a autoridade para colocar mensagens na fila.. A chamada poderá ser bem-sucedida somente se o *AlternateUserId* estiver autorizado a abrir a fila com as opções especificadas, independentemente de o identificador de usuário sob o qual o aplicativo está em execução estar autorizado a fazer isso ou não. (Isso não se aplica às opções de contexto especificadas, no entanto, que são sempre verificadas com relação ao identificador de usuário sob o qual o aplicativo está em execução.)..

Esta opção é válida apenas com a chamada MQPUT1 ..

### **MQPMO\_FAIL\_IF QUIESCING**

Esta opção força a chamada MQPUT ou MQPUT1 a falhar se o gerenciador de filas estiver no estado quiesce.

No z/OS, essa opção também força a chamada MQPUT ou MQPUT1 a falhar se a conexão (para um aplicativo CICS ou IMS) estiver no estado quiesce.

A chamada retorna o código de conclusão MQCC\_FAILED com o código de razão MQRC\_Q\_MGR QUIESCING ou MQRC\_CONNECTION QUIESCING

### **MQPMO\_RESOLVE\_LOCAL\_Q**

Use essa opção para preencher *ResolvedQName* na estrutura MQPMO com o nome da fila local na qual a mensagem é colocada e *ResolvedQMgrName* com o nome do gerenciador de fila local que hospeda a fila local Para obter mais informações sobre MQPMO\_RESOLVE\_LOCAL\_Q, consulte o tópico [MQOO\\_RESOLVE\\_LOCAL\\_Q](#).

Se você estiver autorizado a colocar em uma fila, terá a autoridade necessária para especificar essa sinalização na chamada MQPUT; nenhuma autoridade especial é necessária.

**Opção padrão** Se você não precisar de nenhuma das opções descritas, use a seguinte opção:

### **MQPMO\_NONE**

Use esse valor para indicar que nenhuma outra opção foi especificada. Todas as opções assumem seus valores-padrão. MQPMO\_NONE é definido para auxiliar a documentação do programa; não se pretende que esta opção seja usada com qualquer outra, mas como seu valor é zero, tal uso não pode ser detectado...

MQPMO\_NONE é um campo de entrada O valor inicial do campo *Options* é MQPMO\_NONE.

### **Tempo limite (MQLONG) para MQPMO**

Este é um campo reservado; seu valor não é significativo. O valor inicial desse campo é -1.

### **Contexto (MQHOBJ) para MQPMO**

Se MQPMO\_PASS\_IDENTITY\_CONTEXT ou MQPMO\_PASS\_ALL\_CONTEXT for especificado, esse campo deverá conter a manipulação de fila de entrada a partir da qual as informações de contexto a serem associadas à mensagem que está sendo colocada serão obtidas

Se nem MQPMO\_PASS\_IDENTITY\_CONTEXT nem MQPMO\_PASS\_ALL\_CONTEXT for especificado, esse campo será ignorado

Este é um campo de entrada. O valor inicial desse campo é 0.

### **KnownDestContagem (MQLONG) para MQPMO**

Este é o número de mensagens que a chamada MQPUT ou MQPUT1 atual enviou com sucesso para filas na lista de distribuição que são filas locais. A contagem não inclui mensagens enviadas para filas que são resolvidas para filas remotas (mesmo que uma fila de transmissão local seja usada inicialmente para armazenar a mensagem) Esse campo também é configurado ao colocar uma mensagem em uma única fila que não esteja em uma lista de distribuição

Esse é um campo de saída. O valor inicial desse campo é 0. Esse campo não será configurado se *Version* for menor que MQPMO\_VERSION\_1.

Esse campo é indefinido no z/OS porque as listas de distribuições não são suportadas

### **UnknownDest(MQLONG) para MQPMO**

Este é o número de mensagens que a chamada MQPUT ou MQPUT1 atual enviou com êxito para filas na lista de distribuição que são resolvidas para filas remotas. As mensagens que o gerenciador de filas retém temporariamente na lista de distribuição contam como o número de destinos individuais que essas listas de distribuição contêm. Esse campo também é configurado ao colocar uma mensagem em uma única fila que não esteja em uma lista de distribuição

Esse é um campo de saída. O valor inicial desse campo é 0. Esse campo não será configurado se *Version* for menor que MQPMO\_VERSION\_1.

Esse campo é indefinido no z/OS porque as listas de distribuições não são suportadas

### **InvalidDestContagem (MQLONG) para MQPMO**

Este é o número de mensagens que não puderam ser enviadas para as filas na lista de distribuição A contagem inclui filas que falharam ao abrir, bem como filas que foram abertas com êxito, mas para as quais a operação put falhou.. Esse campo também é configurado ao colocar uma mensagem em uma única fila que não esteja em uma lista de distribuição

**Nota:** Esse campo será configurado se o parâmetro **CompCode** na chamada MQPUT ou MQPUT1 for MQCC\_OK ou MQCC\_WARNING; ele poderá ser configurado se o parâmetro **CompCode** for MQCC\_FAILED, mas não dependerem disso no código do aplicativo.

Esse é um campo de saída. O valor inicial desse campo é 0. Esse campo não será configurado se *Version* for menor que MQPMO\_VERSION\_1.

Esse campo é indefinido no z/OS porque as listas de distribuições não são suportadas

### **ResolvedQName (MQCHAR48) para MQPMO**

Este é o nome da fila de destino após a resolução do nome ter sido executada pelo gerenciador de fila local O nome retornado é o nome de uma fila existente no gerenciador de filas identificado por *ResolvedQMGrName*.

Um valor não em branco é retornado somente se o objeto for uma fila única; se o objeto for uma lista de distribuição ou um tópico, o valor retornado será indefinido.

Esse é um campo de saída. O comprimento desse campo é fornecido por MQ\_Q\_NAME\_LENGTH O valor inicial deste campo é a cadeia nula em C e 48 caracteres em branco em outras linguagens de programação.

### ***ResolvedQMgrNome (MQCHAR48) para MQPMO***

Este é o nome do gerenciador de fila de destino após a resolução do nome ter sido executada pelo gerenciador de fila local. O nome retornado é o nome do gerenciador de filas que possui a fila identificada por *ResolvedQName* e pode ser o nome do gerenciador de fila local.

Se *ResolvedQName* for uma fila compartilhada pertencente ao grupo de filas compartilhadas ao qual o gerenciador de filas locais pertence, *ResolvedQMgrName* será o nome do grupo de filas compartilhadas. Se a fila for de propriedade de algum outro grupo de filas compartilhadas, o *ResolvedQName* poderá ser o nome do grupo de filas compartilhadas ou o nome de um gerenciador de filas que é um membro do grupo de filas compartilhadas (a natureza do valor retornado é determinada pelas definições de fila que existem no gerenciador de fila local).

Um valor não em branco é retornado somente se o objeto for uma fila única; se o objeto for uma lista de distribuição ou um tópico, o valor retornado será indefinido.

Esse é um campo de saída. O comprimento desse campo é fornecido por `MQ_Q_MGR_NAME_LENGTH`. O valor inicial deste campo é a cadeia nula em C e 48 caracteres em branco em outras linguagens de programação.

### ***RecsPresent (MQLONG) para MQPMO***

Esse é o número de registros de mensagens MQPMR put ou de registros de resposta MQRR que foram fornecidos pelo aplicativo. Esse número pode ser maior que zero somente se a mensagem estiver sendo colocada em uma lista de distribuição.. Registros de mensagens put e registros de resposta são opcionais; o aplicativo não precisa fornecer nenhum registro ou pode optar por fornecer registros de apenas um tipo. Entretanto, se o aplicativo fornecer registros de ambos os tipos, ele deverá fornecer registros *RecsPresent* de cada tipo.

O valor de *RecsPresent* não precisa ser igual ao número de destinos na lista de distribuição. Se muitos registros forem fornecidos, o excesso não será usado; se poucos registros forem fornecidos, os valores padrão serão usados para as propriedades de mensagem para os destinos que não possuem registros de mensagens de colocação (consulte *PutMsgRecOffset*).

Se *RecsPresent* for menor que zero ou maior que zero, mas a mensagem não estiver sendo colocada em uma lista de distribuição, a chamada falhará com o código de razão `MQRC_RECS_PRESENT_ERROR`.

Este é um campo de entrada. O valor inicial desse campo é 0. Esse campo será ignorado se *Version* for menor que `MQPMO_VERSION_2`.

### ***PutMsgRecFields (MQLONG) para MQPMO***

Este campo contém sinalizadores que indicam quais campos MQPMR estão presentes nos registros de mensagens de colocação fornecidos pelo aplicativo. Use *PutMsgRecFields* apenas quando a mensagem estiver sendo colocada em uma lista de distribuição. O campo será ignorado se *RecsPresent* for zero ou ambos *PutMsgRecOffset* e *PutMsgRecPtr* forem zero.

Para campos que estão presentes, o gerenciador de filas usa para cada destino os valores dos campos no registro de mensagem put correspondente. Para campos ausentes, o gerenciador de filas usa os valores da estrutura MQMD.

Use um ou mais dos sinalizadores a seguir para indicar quais campos estão presentes nos registros de mensagens de colocação:

#### **MQPMRF\_MSG\_ID**

O campo identificador de mensagem está presente.

#### **MQPMRF\_CORREL\_ID**

O campo do identificador de correlação está presente.

#### **MQPMRF\_GROUP\_ID**

O campo identificador de grupo está presente.

#### **MQPMRF\_FEEDBACK**

Campo de feedback está presente.



## **MQPMRF\_ACCOUNTING\_TOKEN**

O campo token de contabilidade está presente.

Se você especificar essa sinalização, especifique MQPMO\_SET\_IDENTITY\_CONTEXT ou MQPMO\_SET\_ALL\_CONTEXT no campo *Options* ; se essa condição não for satisfeita, a chamada falhará com o código de razão MQRC\_PMO\_RECORD\_FLAGS\_ERROR.

Se nenhum campo MQPMR estiver presente, o seguinte poderá ser especificado:

## **MQPMRF\_NONE**

Nenhum campo de registro de mensagem de colocação está presente

Se esse valor for especificado, *RecsPresent* deverá ser zero ou *PutMsgRecOffset* e *PutMsgRecPtr* deverá ser zero.

MQPMRF\_NONE é definido para a documentação do programa de auxílio Não se pretende que esta constante seja usada com qualquer outra, mas como seu valor é zero, tal uso não pode ser detectado.

Se *PutMsgRecFields* contém sinalizadores que não são válidos ou registros de mensagens put são fornecidos, mas *PutMsgRecFields* tem o valor MQPMRF\_NONE, a chamada falha com o código de razão MQRC\_PMO\_RECORD\_FLAGS\_ERROR.

Este é um campo de entrada. O valor inicial deste campo é MQPMRF\_NONE. Esse campo será ignorado se *Version* for menor que MQPMO\_VERSION\_2.

## ***PutMsgRecOffset (MQLONG) para MQPMO***

Este é o deslocamento em bytes do primeiro registro de mensagem MQPMR put a partir do início da estrutura MQPMO. O deslocamento pode ser positivo ou negativo *PutMsgRecOffset* é usado apenas quando a mensagem está sendo colocada em uma lista de distribuição O campo será ignorado se *RecsPresent* for zero.

Quando a mensagem está sendo colocada em uma lista de distribuição, uma matriz de um ou mais registros de mensagens de colocação MQPMR pode ser fornecida para especificar determinadas propriedades da mensagem para cada destino individualmente; essas propriedades são:

- ID da Mensagem
- Identificador de correlação
- Identificador de grupo
- Valor de feedback
- Símbolo de contabilidade

Não é necessário especificar todas essas propriedades, mas qualquer subconjunto escolhido, especifique os campos na ordem correta. Consulte a descrição da estrutura do MQPMR para obter detalhes adicionais

Geralmente, deve haver tantos registros de mensagem put quantos houver registros de objetos especificados pelo MQOD quando a lista de distribuição for aberta; cada registro de mensagem put fornece as propriedades de mensagens para a fila identificada pelo registro de objeto correspondente As filas na lista de distribuição que falham ao abrir ainda devem ter registros de mensagens colocados alocados para eles nas posições apropriadas na matriz, embora as propriedades de mensagem sejam ignoradas neste caso

O número de registros de mensagens put pode ser diferente do número de registros de objetos Se houver menos registros de mensagens put do que registros de objetos, as propriedades de mensagens para os destinos que não possuem registros de mensagens put serão obtidas dos campos correspondentes no MQMD do descritor de mensagens Se houver mais registros de mensagens de colocação do que registros de objetos, o excesso não será usado (embora ainda seja possível acessá-los). Os registros de mensagens put são opcionais, mas se forem fornecidos, deve haver *RecsPresent* deles.

Forneça os registros de mensagens put de uma maneira semelhante aos registros de objetos em MQOD, especificando um deslocamento em *PutMsgRecOffset* ou especificando um endereço em *PutMsgRecPtr* ; para obter detalhes de como fazer isso, consulte o campo *ObjectRecOffset* descrito em “MQOD-Descritor de objetos” na página 497..



Não mais de um de *PutMsgRecOffset* e *PutMsgRecPtr* pode ser usado; a chamada falhará com o código de razão MQRC\_PUT\_MSG\_RECORDS\_ERROR se ambos forem diferentes de zero

Este é um campo de entrada. O valor inicial desse campo é 0. Esse campo será ignorado se *Version* for menor que MQPMO\_VERSION\_2.

### ***ResponseRecOffset (MQLONG) para MQPMO***

Este é o deslocamento em bytes do primeiro registro de resposta MQRR do início da estrutura MQPMO. O deslocamento pode ser positivo ou negativo *ResponseRecOffset* é usado apenas quando a mensagem está sendo colocada em uma lista de distribuição O campo será ignorado se *RecsPresent* for zero.

Ao colocar a mensagem em uma lista de distribuição, é possível fornecer uma matriz de um ou mais registros de respostas MQRR para identificar as filas para as quais a mensagem não foi enviada com êxito (campo *CompCode* em MQRR) e a razão para cada falha (campo *Reason* em MQRR). A mensagem pode não ter sido enviada porque a fila falhou ao abrir ou porque a operação put falhou. O gerenciador de filas configura os registros de respostas apenas quando o resultado da chamada é misto (ou seja, algumas mensagens foram enviadas com sucesso enquanto outras falharam ou todas falharam, mas por razões diferentes); o código de razão MQRC\_MULTIPLE\_REASON da chamada indica esse caso Se o mesmo código de razão se aplicar a todas as filas, esse motivo será retornado no parâmetro **Reason** da chamada MQPUT ou MQPUT1 e os registros de resposta não serão configurados.

Geralmente, há tantos registros de resposta quanto há registros de objetos especificados por MQOD quando a lista de distribuições é aberta; quando necessário, cada registro de resposta é configurado para o código de conclusão e o código de razão para a colocação na fila identificada pelo registro de objeto correspondente As filas na lista de distribuição que falham ao abrir ainda devem ter registros de resposta alocados para elas nas posições apropriadas na matriz, embora elas estejam configuradas para o código de conclusão e o código de razão resultantes da operação aberta, em vez da operação put.

O número de registros de resposta pode ser diferente do número de registros de objeto Se houver menos registros de respostas do que registros de objetos, o aplicativo poderá não ser capaz de identificar todos os destinos para os quais a operação put falhou ou as razões para as falhas Se houver mais registros de resposta do que registros de objeto, o excesso não será usado (embora ainda seja possível acessá-los). Os registros de resposta são opcionais, mas se forem fornecidos, deverá haver *RecsPresent* deles.

Forneça os registros de resposta de maneira semelhante aos registros de objeto em MQOD, especificando um deslocamento em *ResponseRecOffset* ou especificando um endereço em *ResponseRecPtr* ; para obter detalhes de como fazer isso, consulte o campo *ObjectRecOffset* descrito em “MQOD-  
Descritor de objetos” na página 497.. No entanto, use no máximo um de *ResponseRecOffset* e *ResponseRecPtr* ; a chamada falha com o código de razão MQRC\_RESPONSE\_RECORDS\_ERROR se ambos não forem zero.

Para a chamada MQPUT1 , esse campo deve ser zero. Isso ocorre porque as informações de resposta (se solicitadas) são retornadas nos registros de resposta especificados pelo descritor de objeto MQOD.

Este é um campo de entrada. O valor inicial desse campo é 0. Esse campo será ignorado se *Version* for menor que MQPMO\_VERSION\_2.

### ***PutMsgRecPtr (MQPTR) para MQPMO***

Este é o endereço do primeiro registro de mensagem de colocação do MQPMR Use *PutMsgRecPtr* apenas quando a mensagem estiver sendo colocada em uma lista de distribuição O campo será ignorado se *RecsPresent* for zero.

É possível usar *PutMsgRecPtr* ou *PutMsgRecOffset* pode ser usado para especificar os registros de mensagens put, mas não ambos; para obter detalhes, consulte “PutMsgRecOffset (MQLONG) para MQPMO” na página 532. Se você não usar *PutMsgRecPtr*, configure-o para o ponteiro nulo ou bytes nulos.

Este é um campo de entrada. O valor inicial deste campo é o ponteiro nulo nas linguagens de programação que suportam ponteiros e, caso contrário, uma cadeia de bytes all-null. Esse campo será ignorado se *Version* for menor que MQPMO\_VERSION\_2.

**Nota:** Em plataformas em que a linguagem de programação não suporta o tipo de dados do ponteiro, este campo é declarado como uma cadeia de bytes do comprimento apropriado, sendo o valor inicial a cadeia de bytes all-null.

### ***ResponseRecPtr (MQPTR) para MQPMO***

Este é o endereço do primeiro registro de resposta do MQRR *ResponseRecPtr* é usado apenas quando a mensagem está sendo colocada em uma lista de distribuição O campo será ignorado se *RecsPresent* for zero.

Use *ResponseRecPtr* ou *ResponseRecOffset* para especificar os registros de resposta, mas não ambos; para obter detalhes, consulte “[ResponseRecOffset \(MQLONG\) para MQPMO](#)” na página 533 Se você não usar *ResponseRecPtr*, configure-o para o ponteiro nulo ou bytes nulos.

Para a chamada MQPUT1, esse campo deve ser o ponteiro nulo ou bytes nulos. Isso ocorre porque as informações de resposta (se solicitadas) são retornadas nos registros de resposta especificados pelo descritor de objeto MQOD.

Este é um campo de entrada. O valor inicial deste campo é o ponteiro nulo nas linguagens de programação que suportam ponteiros e, caso contrário, uma cadeia de bytes all-null. Esse campo será ignorado se *Version* for menor que MQPMO\_VERSION\_2.

**Nota:** Em plataformas em que a linguagem de programação não suporta o tipo de dados do ponteiro, este campo é declarado como uma cadeia de bytes do comprimento apropriado, sendo o valor inicial a cadeia de bytes all-null.

### ***OriginalMsgHandle (MQHMSG) para MQPMO***

Esta é uma manipulação opcional para uma mensagem Ele pode ter sido recuperado anteriormente de uma fila. O uso desse identificador está sujeito ao valor do campo *Action*; consulte também [NewMsgHandle](#).

O conteúdo da manipulação da mensagem original não será mudado por a chamada MQPUT ou MQPUT1

Este é um campo de entrada. O valor inicial desse campo é MQHM\_NONE. Esse campo será ignorado se a Versão for menor que MQPMO\_VERSION\_3

### ***NewMsgIdentificador (MQHMSG) para MQPMO***

Este é um identificador opcional para a mensagem sendo colocada sujeita ao valor do campo Ação. Ele define as propriedades da mensagem e substitui os valores do *OriginalMsgHandle*, se especificado

No retorno da chamada MQPUT ou MQPUT1, o conteúdo da manipulação reflete a mensagem que foi realmente colocada.

Este é um campo de entrada. O valor inicial desse campo é MQHM\_NONE. Esse campo será ignorado se a Versão for menor que MQPMO\_VERSION\_3

### ***Ação (MQLONG) para MQPMO***

Isso especifica o tipo de put que está sendo executado e o relacionamento entre a mensagem original especificada pelo campo Manipulação OriginalMsg e a nova mensagem especificada pelo campo Manipulação NewMsg. As propriedades da mensagem são escolhidas pelo gerenciador de filas de acordo com o valor da ação especificada.

É possível escolher fornecer o conteúdo do descritor de mensagens usando o parâmetro MsgDesc nas chamadas MQPUT ou MQPUT1. Como alternativa, é possível não fornecer o parâmetro MsgDesc ou especificar que ele é somente de saída, incluindo MQPMO\_MD\_FOR\_OUTPUT\_ONLY no campo Opções da estrutura MQPMO.

Se o parâmetro `MsgDesc` não for fornecido, ou se for especificado para ser somente de saída, o descritor de mensagem para a nova mensagem será preenchido a partir dos campos de manipulação de mensagens do MQPMO, de acordo com as regras descritas neste tópico

A configuração de contexto e as atividades de transmissão descritas em Informações de contexto de controle entram em vigor após o descritor de mensagens ter sido composto

Se um valor de ação incorreto for especificado, a chamada falhará com o código de razão `MQRC_ACTION_ERROR`.

Qualquer uma das seguintes ações pode ser especificada:

#### **MQACTP\_NEW**

Uma nova mensagem está sendo colocada, e nenhuma relação com uma mensagem anterior está sendo especificada pelo programa. O descritor de mensagens é composto da seguinte forma:

- Se uma chamada `MsgDesc` for fornecida na chamada `MQPUT` ou `MQPUT1` e `MQPMO_MD_FOR_OUTPUT_ONLY` não estiver no `MQPMO MQPMO.Options`, isso é usado como o descritor da mensagem não modificado
- Se um `MsgDesc` não for fornecido, ou `MQPMO_MD_FOR_OUTPUT_ONLY` estiver no `MQPMO MQPMO.Options`, em seguida, o gerenciador de filas gerará o descritor de mensagens usando uma combinação de propriedades do identificador `OriginalMsge` do identificador `NewMsg`. Quaisquer campos do descritor de mensagem explicitamente configurados no novo identificador de mensagem têm precedência sobre aqueles no identificador de mensagem original

Os dados da mensagem são obtidos do parâmetro `MQPUT` ou `MQPUT1 Buffer`.

#### **MQACTP\_FORWARD**

Uma mensagem recuperada anteriormente está sendo encaminhada.. O identificador de mensagem original especifica a mensagem que foi recuperada anteriormente

O novo identificador de mensagens especifica quaisquer modificações nas propriedades (incluindo qualquer no descritor de mensagens) no identificador de mensagens original.

O descritor de mensagens é composto da seguinte forma:

- Se uma chamada `MsgDesc` for fornecida na chamada `MQPUT` ou `MQPUT1` e `MQPMO_MD_FOR_OUTPUT_ONLY` não estiver no `MQPMO MQPMO.Options`, isso é usado como o descritor da mensagem não modificado
- Se um `MsgDesc` não for fornecido, ou `MQPMO_MD_FOR_OUTPUT_ONLY` estiver no `MQPMO MQPMO.Options`, em seguida, o gerenciador de filas gerará o descritor de mensagens usando uma combinação de propriedades do identificador `OriginalMsge` do identificador `NewMsg`. Quaisquer campos do descritor de mensagem explicitamente configurados no novo identificador de mensagem têm precedência sobre aqueles no identificador de mensagem original
- Se `MQPMO_NEW_MSG_ID` ou `MQPMO_NEW_CORREL_ID` forem especificados no `MQPMO.Options`, então elas são honradas.

As propriedades de mensagens são compostas da seguinte forma:

- Todas as propriedades do identificador de mensagem original que possuem `MQCOPY_FORWARD` no `MQPD MQPD.CopyOptions`
- Todas as propriedades da nova manipulação de mensagem Para cada propriedade no novo identificador de mensagens que possui o mesmo nome que uma propriedade no identificador de mensagens original, o valor é obtido do novo identificador de mensagens... A única exceção a essa regra é o caso especial quando a propriedade na nova manipulação de mensagem tiver o mesmo nome que uma propriedade na manipulação de mensagens original, mas o valor da propriedade for nulo.. Nesse caso, a propriedade é removida da mensagem

Os dados da mensagem a serem encaminhados são obtidos do parâmetro `MQPUT` ou `MQPUT1 Buffer`.

#### **MQACTP\_REPLY**

Uma resposta está sendo feita para uma mensagem recuperada anteriormente O identificador de mensagem original especifica a mensagem que foi recuperada anteriormente

O novo identificador de mensagens especifica quaisquer modificações nas propriedades (incluindo qualquer no descritor de mensagens) no identificador de mensagens original.

O descritor de mensagens é composto da seguinte forma:

- Se uma chamada MsgDesc for fornecida na chamada MQPUT ou MQPUT1 e MQPMO\_MD\_FOR\_OUTPUT\_ONLY não estiver no MQPMO MQPMO.Options, isso é usado como o descritor da mensagem não modificado
- Se um MsgDesc não for fornecido, ou MQPMO\_MD\_FOR\_OUTPUT\_ONLY estiver no MQPMO MQPMO.Options, em seguida, os campos do descritor de mensagens iniciais são escolhidos da seguinte forma:

<i>Tabela 509. Transformação de manipulação de mensagem de resposta..</i>	
<b>Campo no MQMD</b>	<b>Valor Usado</b>
Relatório	Se MQRO_PASS_DISCARD_AND_EXPIRE e MQRO_DISCARD_MSG são configurados: MQRO_DISCARD_MSG caso contrário MQRO_NONE
MsgType	MQMT_REPLY
Expiração	Se MQRO_PASS_DISCARD_AND_EXPIRE é configurado:: Copiado da mensagem de entrada caso contrário MQEI_UNLIMITED
Feedback	MQFB_NONE
MsgId	Se MQPMO_NEW_MSG_ID estiver configurado: Um novo identificador de mensagem é gerado caso contrário, se MQRO_PASS_MSG_ID estiver configurado: Copiado da mensagem de entrada caso contrário MQMI_NONE
CorrelId	Se MQPMO_NEW_CORREL_ID for configurado: Um novo identificador de correlação é gerado se MQRO_COPY_MSG_ID_TO_CORREL_ID estiver configurado: Copiado do campo MsgId do mensagem de entrada se MQRO_PASS_CORREL_ID estiver configurado: Copiado do campo CorrelId do mensagem de entrada caso contrário MQCI_NONE
BackoutCount	0
ReplyToQ	Espaços em branco
ReplyToQMgr	Espaços em branco
GroupId	MQGI_NONE

<i>Tabela 509. Transformação de manipulação de mensagem de resposta.. (continuação)</i>	
<b>Campo no MQMD</b>	<b>Valor Usado</b>
MsgSeqNumber	1
Offset	0
MsgFlags	MQMF_NONE
OriginalLength	MQOL_UNDEFINED

- O descritor de mensagens é então modificado pelo novo identificador de mensagens-quaisquer campos do descritor de mensagens explicitamente configurados como propriedades no novo identificador de mensagens têm precedência sobre os campos do descritor de mensagens, conforme descrito anteriormente.

As propriedades de mensagens são compostas da seguinte forma:

- Todas as propriedades do identificador de mensagem original que possuem MQCOPY\_REPLY no MQPD do MQPD.CopyOptions
- Todas as propriedades da nova manipulação de mensagem Para cada propriedade no novo identificador de mensagens que possui o mesmo nome que uma propriedade no identificador de mensagens original, o valor é obtido do novo identificador de mensagens... A única exceção a essa regra é o caso especial quando a propriedade na nova manipulação de mensagem tiver o mesmo nome que uma propriedade na manipulação de mensagens original, mas o valor da propriedade for nulo.. Nesse caso, a propriedade é removida da mensagem

Os dados da mensagem a serem encaminhados são obtidos do parâmetro Buffer MQPUT/MQPUT1 .

#### **MQACTP\_REPORT**

Um relatório está sendo gerado como um resultado de uma mensagem recuperada anteriormente O identificador de mensagem original especifica a mensagem que está causando a geração do relatório.

O novo identificador de mensagens especifica quaisquer modificações nas propriedades (incluindo qualquer no descritor de mensagens) no identificador de mensagens original.

O descritor de mensagens é composto da seguinte forma:

- Se uma chamada MsgDesc for fornecida na chamada MQPUT ou MQPUT1 e MQPMO\_MD\_FOR\_OUTPUT\_ONLY não estiver no MQPMO MQPMO.Options, isso é usado como o descritor da mensagem não modificado
- Se um MsgDesc não for fornecido, ou MQPMO\_MD\_FOR\_OUTPUT\_ONLY estiver no MQPMO MQPMO.Options , em seguida, os campos do descritor de mensagens iniciais são escolhidos da seguinte forma:

<i>Tabela 510. Transformação de manipulação de mensagem de relatório..</i>	
<b>Campo no MQMD</b>	<b>Valor Usado</b>
Relatório	Se MQRO_PASS_DISCARD_AND_EXPIRE e MQRO_DISCARD_MSG são configuradas: MQRO_DISCARD_MSG caso contrário MQRO_NONE
MsgType	MQMT_REPORT
Expiração	Se MQRO_PASS_DISCARD_AND_EXPIRE é configurado:: Copiado da mensagem de entrada caso contrário MQEI_UNLIMITED

Tabela 510. Transformação de manipulação de mensagem de relatório.. (continuação)

Campo no MQMD	Valor Usado
MsgId	Se MQPMO_NEW_MSG_ID estiver configurado: Um novo identificador de mensagem é gerado caso contrário, se MQRO_PASS_MSG_ID estiver configurado: Copiado da mensagem de entrada caso contrário MQMI_NONE
CorrelId	Se MQPMO_NEW_CORREL_ID for configurado: Um novo identificador de correlação é gerado se MQRO_COPY_MSG_ID_TO_CORREL_ID estiver configurado: Copiado do campo MsgId do mensagem de entrada se MQRO_PASS_CORREL_ID estiver configurado: Copiado do campo CorrelId do mensagem de entrada caso contrário MQCI_NONE
BackoutCount	0
ReplyToQ	Espaços em branco
ReplyToQMgr	Espaços em branco
OriginalLength	Configure como <i>BufferLength</i>

- O descritor de mensagens é então modificado pelo novo identificador de mensagens-quaisquer campos do descritor de mensagens explicitamente configurados como propriedades no novo identificador de mensagens têm precedência sobre os campos do descritor de mensagens, conforme descrito anteriormente.

As propriedades de mensagens são compostas da seguinte forma:

- Todas as propriedades do identificador de mensagem original que possuem MQCOPY\_REPORT no MQPD MQPD.CopyOptions
- Todas as propriedades da nova manipulação de mensagem Para cada propriedade no novo identificador de mensagens que possui o mesmo nome que uma propriedade no identificador de mensagens original, o valor é obtido do novo identificador de mensagens... A única exceção a essa regra é o caso especial quando a propriedade na nova manipulação de mensagem tiver o mesmo nome que uma propriedade na manipulação de mensagens original, mas o valor da propriedade for nulo.. Nesse caso, a propriedade é removida da mensagem

O campo Feedback no MQMD resultante representa o relatório que será gerado. Um valor de feedback de MQFB\_NONE faz com que a chamada MQPUT ou MQPUT1 falhe com o código de razão MQRC\_FEEDBACK\_ERROR.

Para escolher os dados do usuário da mensagem de relatório, o IBM MQ consulta os campos Relatório e Feedback no MQMD resultante e os parâmetros Buffer e BufferLength da chamada MQPUT ou MQPUT1 .

- Se Feedback for MQFB\_COA, MQFB\_COD ou MQFB\_EXPIRATION, o valor do Relatório será inspecionado.
- Se qualquer um dos casos a seguir for true, os dados da mensagem integral do Buffer para um comprimento de BufferLength serão usados

- O feedback é MQFB\_EXPIRATION e o Relatório contém MQRO\_EXPIRATION\_WITH\_FULL\_DATA
- O feedback é MQFB\_COD e Report contém MQRO\_COD\_WITH\_FULL\_DATA
- O feedback é MQFB\_COA e o Relatório contém MQRO\_COA\_WITH\_FULL\_DATA
- Se qualquer um dos casos a seguir for true, os primeiros 100 bytes da mensagem (ou BufferLength se for menor que 100) do Buffer serão usados
  - O feedback é MQFB\_EXPIRATION e o Relatório contém MQRO\_EXPIRATION\_WITH\_DATA
  - O feedback é MQFB\_COD e o relatório contém MQRO\_COD\_WITH\_DATA
  - O feedback é MQFB\_COA e o relatório contém MQRO\_COA\_WITH\_DATA
- Se o Feedback for MQFB\_EXPIRATION, MQFB\_COD ou MQFB\_COA e o Relatório não contiver as opções \*\_WITH\_FULL\_DATA ou \*\_WITH\_DATA relevantes para esse valor de Feedback, nenhum dado do usuário será incluído com a mensagem.
- Se Feedback usa um valor diferente daqueles listados acima, então Buffer e BufferLength são usados normalmente.

A derivação dos dados do usuário descritos na lista anterior também é mostrada na tabela a seguir:

<i>Tabela 511. Origem de dados do usuário</i>			
	<b>MQFB_COA</b>	<b>MQFB_COD</b>	<b>MQFB_EXPIRATION</b>
<b>MQRO_EXPIRATION_WITH_FULL_DATA</b>	Nenhum	Nenhum	Buffer (Bufferlength)
<b>MQRO_COD_WITH_FULL_DATA</b>	Nenhum	Buffer (Bufferlength)	Nenhum
<b>MQRO_COA_WITH_FULL_DATA</b>	Buffer (Bufferlength)	Nenhum	Nenhum
<b>MQRO_EXPIRATION_WITH_DATA</b>	Nenhum	Nenhum	Buffer (primeiros 100 bytes)
<b>MQRO_COD_WITH_DATA</b>	Nenhum	Buffer (primeiros 100 bytes)	Nenhum
<b>MQRO_COA_WITH_DATA</b>	Buffer (primeiros 100 bytes)	Nenhum	Nenhum

### **PubLevel (MQLONG) para MQPMO**

O valor inicial deste campo é de 9. O nível de assinatura destinado por esta publicação Somente as assinaturas com o SubLevel mais alto menor ou igual a esse valor recebem essa publicação... Este valor deve estar no intervalo de zero a 9; zero é o nível mais baixo. No entanto, se uma publicação tiver sido retida, ela não estará mais disponível para assinantes em níveis mais altos porque ela será publicada novamente em PubLevel 1.

Para obter informações, consulte [Interceptando publicações..](#)

### **MQPMR-Registro de mensagem de colocação**

Use a estrutura MQPMR para especificar várias propriedades de mensagens para um único destino ao colocar uma mensagem em uma lista de distribuições MQPMR é uma estrutura de entrada / saída para chamadas MQPUT e MQPUT1 .

### **Disponibilidade**

A estrutura MQPMR está disponível nas plataformas a seguir:

-  AIX

-  IBM i
-  Linux
-  Windows

e para clientes IBM MQ conectados a esses sistemas.

## Conjunto de caracteres e codificação

Os dados em MQPMR devem estar no conjunto de caracteres fornecido pelo atributo do gerenciador de filas do **CodedCharSetId** e pela codificação do gerenciador de fila local fornecido por MQENC\_NATIVE. No entanto, se o aplicativo estiver em execução como um cliente MQ, a estrutura deverá estar no conjunto de caracteres e na codificação do cliente..

## Uso

Fornecendo uma matriz dessas estruturas na chamada MQPUT ou MQPUT1, é possível especificar valores diferentes para cada fila de destino em uma lista de distribuição. Alguns dos campos são apenas de entrada, outros são de entrada/saída.

**Nota:** Essa estrutura é incomum porque não possui um layout fixo. Os campos nessa estrutura são opcionais e a presença ou ausência de cada campo é indicada pelas sinalizações no campo *PutMsgRecFields* em MQPMO. Os campos que estão presentes **devem ocorrer na seguinte ordem** :

- *MsgId*
- *CorrelId*
- *GroupId*
- *Feedback*
- *AccountingToken*

Os campos ausentes não ocupam espaço no registro.

Como MQPMR não tem um layout fixo, nenhuma definição dele é fornecida no cabeçalho, arquivos COPY e INCLUDE para as linguagens de programação suportadas. O programador de aplicativos deve criar uma declaração contendo os campos que são necessários para o aplicativo e configurar os sinalizadores em *PutMsgRecFields* para indicar os campos que estão presentes.

## Campos

Não há valores iniciais definidos para essa estrutura, pois nenhuma declaração de estrutura é fornecida nos arquivos de cabeçalho COPY e INCLUDE para as linguagens de programação suportadas. As declarações de amostra mostram como declarar a estrutura se todos os campos forem necessários..

<i>Tabela 512. Campos em MQPMR</i>	
<b>Nome de Campo</b>	<b>Descrição do Campo</b>
<u>MsgId</u>	ID da Mensagem
<u>CorrelId</u>	Identificador de correlação
<u>GroupId</u>	Identificador de grupo
<u>Feedback</u>	Feedback ou código de razão
<u>AccountingToken</u>	Símbolo de contabilidade



## Declarações de idiomas

### Declaração C para MQPMR

```
typedef struct tagMQPMR MQPMR;
struct tagMQPMR {
    MQBYTE24 MsgId;           /* Message identifier */
    MQBYTE24 CorrelId;       /* Correlation identifier */
    MQBYTE24 GroupId;        /* Group identifier */
    MQLONG Feedback;         /* Feedback or reason code */
    MQBYTE32 AccountingToken; /* Accounting token */
};
```

### Declaração COBOL para MQPMR

```
** MQPMR structure
10 MQPMR.
** Message identifier
15 MQPMR-MSGID PIC X(24).
** Correlation identifier
15 MQPMR-CORRELID PIC X(24).
** Group identifier
15 MQPMR-GROUPID PIC X(24).
** Feedback or reason code
15 MQPMR-FEEDBACK PIC S9(9) BINARY.
** Accounting token
15 MQPMR-ACCOUNTINGTOKEN PIC X(32).
```

### Declaração PL/I para MQPMR

```
dcl
1 MQPMR based,
3 MsgId char(24), /* Message identifier */
3 CorrelId char(24), /* Correlation identifier */
3 GroupId char(24), /* Group identifier */
3 Feedback fixed bin(31), /* Feedback or reason code */
3 AccountingToken char(32); /* Accounting token */
```

### Declaração Visual Basic para MQPMR

```
Type MQPMR
MsgId As MQBYTE24 'Message identifier'
CorrelId As MQBYTE24 'Correlation identifier'
GroupId As MQBYTE24 'Group identifier'
Feedback As Long 'Feedback or reason code'
AccountingToken As MQBYTE32 'Accounting token'
End Type
```

## ***MsgId (MQBYTE24) para MQPMR***

Este é o identificador de mensagem a ser usado para a mensagem enviada para a fila com um nome especificado pelo elemento correspondente na matriz de estruturas MQOR fornecidas na chamada MQOPEN ou MQPUT1. Ele é processado da mesma forma que o campo *MsgId* no MQMD para uma colocação em uma única fila.

Se esse campo não estiver presente no registro MQPMR ou houver menos registros MQPMR do que destinos, o valor em MQMD será usado para os destinos que não tiverem um registro MQPMR contendo um campo *MsgId*. Se esse valor for MQMI\_NONE, um novo identificador de mensagem será gerado para cada desses destinos (ou seja, nenhum desses destinos possui o mesmo identificador de mensagens).

Se MQPMO\_NEW\_MSG\_ID for especificado, novos identificadores de mensagens serão gerados para todos os destinos na lista de distribuição, independentemente se eles tiverem registros MQPMR. Isso é diferente da maneira como MQPMO\_NEW\_CORREL\_ID é processado (consulte o campo *CorrelId*).

Este é um campo de entrada/saída

### ***CorrelId (MQBYTE24) para MQPMR***

Este é o identificador de correlação a ser usado para a mensagem enviada para a fila com um nome especificado pelo elemento correspondente na matriz de estruturas MQOR fornecidas na chamada MQOPEN ou MQPUT1 . Ele é processado da mesma forma que o campo *CorrelId* no MQMD para uma colocação em uma única fila

Se esse campo não estiver presente no registro MQPMR ou houver menos registros MQPMR do que destinos, o valor em MQMD será usado para os destinos que não tiverem um registro MQPMR contendo um campo *CorrelId* .

Se MQPMO\_NEW\_CORREL\_ID for especificado, um novo identificador de correlação *único* será gerado e usado para todos os destinos na lista de distribuições, independentemente de eles terem registros MQPMR. Isso é diferente da maneira como MQPMO\_NEW\_MSG\_ID é processado (consulte o campo *MsgId* ).

Este é um campo de entrada/saída

### ***GroupId (MQBYTE24) para MQPMR.***

GroupId é o identificador de grupo a ser usado para a mensagem enviada para a fila com o nome especificado pelo elemento correspondente na matriz de estruturas MQOR fornecidas na chamada MQOPEN ou MQPUT1 . Ele é processado da mesma forma que o campo *GroupId* no MQMD para uma colocação em uma única fila

Se esse campo não estiver presente no registro MQPMR ou houver menos registros MQPMR do que destinos, o valor em MQMD será usado para os destinos que não tiverem um registro MQPMR contendo um campo *GroupId* . O valor é processado conforme documentado em [Ordem física em uma fila](#), mas com as diferenças a seguir:

- GroupId é criado a partir do QMName e um registro de data e hora. Portanto, manter um GroupId exclusivo manter nomes do gerenciador de filas exclusivos também. Além disso, não configure os clocks de volta na máquina de gerenciadores de filas
- Nos casos em que um novo identificador de grupo seria usado, o gerenciador de filas gerará um identificador de grupo diferente para cada destino (ou seja, nenhum dos dois destinos possui o mesmo identificador de grupo).
- Nos casos em que o valor no campo seria usado, a chamada falhará com o código de razão MQRC\_GROUP\_ID\_ERROR

Este é um campo de entrada/saída

### ***Feedback (MQLONG) para MQPMR***

Esse é o código de feedback a ser usado para a mensagem enviada para a fila com o nome especificado pelo elemento correspondente na matriz de estruturas MQOR fornecidas na chamada MQOPEN ou MQPUT1 . Ele é processado da mesma forma que o campo *Feedback* no MQMD para uma colocação em uma única fila

Se esse campo não estiver presente, o valor em MQMD será usado.

Este é um campo de entrada.

### ***AccountingToken (MQBYTE32) para MQPMR***

Esse é o token de contabilidade a ser usado para a mensagem enviada para a fila com o nome especificado pelo elemento correspondente na matriz de estruturas MQOR fornecidas na chamada MQOPEN ou MQPUT1 . Ele é processado da mesma forma que o campo *AccountingToken* no MQMD para uma colocação em uma única fila Consulte a descrição de *AccountingToken* em [“MQMD - descritor de mensagem”](#) na página 435 para obter informações sobre o conteúdo desse campo

Se esse campo não estiver presente, o valor em MQMD será usado.

Este é um campo de entrada.

## MQRFH - Regras e formatação de cabeçalho

A estrutura MQRFH define o layout das regras e o cabeçalho de formatação Use este cabeçalho para enviar dados de sequência na forma de pares nome-valor..

### Disponibilidade

Todos os sistemas IBM MQ , além de IBM MQ MQI clients conectados a esses sistemas

### Nome do Formato

MQFMT\_RF\_HEADER

### Conjunto de caracteres e codificação

Os campos na estrutura MQRFH (incluindo *NameValueString*) estão no conjunto de caracteres e na codificação fornecidos pelos campos *CodedCharSetId* e *Encoding* na estrutura do cabeçalho que precede o MQRFH ou por esses campos na estrutura MQMD se o MQRFH estiver no início dos dados da mensagem do aplicativo.

O conjunto de caracteres deve ser um que tenha caracteres de byte único para os caracteres válidos em nomes de filas.

### Campos

**Nota:** Na tabela a seguir, os campos são agrupados por uso e não alfabeticamente. Os tópicos filhos seguem a mesma sequência.

Tabela 513. Campos em MQRFH para MQRFH		
Nome e descrição do campo	Nome da constante	Valor inicial (se houver) da constante
<u>StrucId</u> (identificador de estrutura)	MQRFH_STRUC_ID	'RFH→'
<u>Versão</u> (número da versão da estrutura).	MQRFH_VERSION_1	1
<u>StrucLength</u> (comprimento em bytes da estrutura MQRFH)	MQRFH_STRUC_LEN GH_FIXED	32
<u>Codificação</u> (codificação numérica dos dados que seguem <i>NameValueString</i> ).	MQENC_NATIVE	Depende do ambiente
<u>CodedCharSetId</u> (especifica o identificador do conjunto de caractere dos dados que seguem <i>NameValueString</i> )	MQCCSI_UNDEFINED	0
<u>Formato</u> (nome do formato dos dados após <i>NameValueString</i> )	MQFMT_NONE	Espaços em branco
<u>Sinalizadores</u> (sinalizadores)	MQRFH_NONE	0
<u>NameValueString</u> (sequência de caracteres de comprimento variável contendo pares nome-valor)	Nenhum	Nenhum

Tabela 513. Campos em MQRFH para MQRFH (continuação)

Nome e descrição do campo	Nome da constante	Valor inicial (se houver) da constante
<b>Notas:</b>		
<ol style="list-style-type: none"> <li>1. O símbolo <code>\n</code> representa um único caractere em branco.</li> <li>2. Na linguagem de programação C, a variável macro <code>MQRFH_DEFAULT</code> contém os valores que são listados na tabela. Ele pode ser usado da seguinte maneira para fornecer valores iniciais para os campos na estrutura: <pre> MQRFH MyRFH = {MQRFH_DEFAULT}; </pre> </li> </ol>		

## Declarações de idiomas

### Declaração C para MQRFH

```

typedef struct tagMQRFH MQRFH;
struct tagMQRFH {
    MQCHAR4  StrucId;          /* Structure identifier */
    MQLONG   Version;         /* Structure version number */
    MQLONG   StrucLength;     /* Total length of MQRFH including
                               NameValueString */
    MQLONG   Encoding;       /* Numeric encoding of data that follows
                               NameValueString */
    MQLONG   CodedCharSetId; /* Character set identifier of data that
                               follows NameValueString */
    MQCHAR8  Format;          /* Format name of data that follows
                               NameValueString */
    MQLONG   Flags;          /* Flags */
};

```

### Declaração COBOL para MQRFH

```

** MQRFH structure
10 MQRFH.
** Structure identifier
15 MQRFH-STRUCID PIC X(4).
** Structure version number
15 MQRFH-VERSION PIC S9(9) BINARY.
** Total length of MQRFH including NAMEVALUESTRING
15 MQRFH-STRUCLength PIC S9(9) BINARY.
** Numeric encoding of data that follows NAMEVALUESTRING
15 MQRFH-ENCODING PIC S9(9) BINARY.
** Character set identifier of data that follows NAMEVALUESTRING
15 MQRFH-CODEDCHARSETID PIC S9(9) BINARY.
** Format name of data that follows NAMEVALUESTRING
15 MQRFH-FORMAT PIC X(8).
** Flags
15 MQRFH-FLAGS PIC S9(9) BINARY.

```

### Declaração PL/I para MQRFH

```

dcl
1 MQRFH based,
3 StrucId char(4), /* Structure identifier */
3 Version fixed bin(31), /* Structure version number */
3 StrucLength fixed bin(31), /* Total length of MQRFH including
                               NameValueString */
3 Encoding fixed bin(31), /* Numeric encoding of data that
                               follows NameValueString */
3 CodedCharSetId fixed bin(31), /* Character set identifier of data
                               that follows NameValueString */
3 Format char(8), /* Format name of data that follows
                               NameValueString */
3 Flags fixed bin(31); /* Flags */

```

## Declaração do High Level Assembler para MQRFH

```
MQRFH          DSECT
MQRFH_STRUCID  DS   CL4  Structure identifier
MQRFH_VERSION  DS   F    Structure version number
MQRFH_STRUCLength DS   F    Total length of MQRFH including
*              NAMEVALUESTRING
MQRFH_ENCODING DS   F    Numeric encoding of data that follows
*              NAMEVALUESTRING
MQRFH_CODEDCHARSETID DS   F    Character set identifier of data that
*              follows NAMEVALUESTRING
MQRFH_FORMAT   DS   CL8  Format name of data that follows
*              NAMEVALUESTRING
MQRFH_FLAGS    DS   F    Flags
*
MQRFH_LENGTH   EQU  *-MQRFH
                ORG  MQRFH
MQRFH_AREA     DS   CL(MQRFH_LENGTH)
```

## Declaração Visual Basic para MQRFH

```
Type MQRFH
  StrucId      As String*4 'Structure identifier'
  Version      As Long     'Structure version number'
  StrucLength  As Long     'Total length of MQRFH including'
                'NameValueString'
  Encoding     As Long     'Numeric encoding of data that follows'
                'NameValueString'
  CodedCharSetId As Long   'Character set identifier of data that'
                'follows NameValueString'
  Format        As String*8 'Format name of data that follows'
                'NameValueString'
  Flags        As Long     'Flags'
End Type
```

### **StrucId (MQRFH) para MQRFH**

Este é o identificador de estrutura da estrutura do cabeçalho de regras e de formatação. Ele é sempre um campo de entrada. Seu valor é MQRFH\_STRUC\_ID.

O valor deve ser:

#### **MQRFH\_STRUC\_ID**

Identificador para a estrutura do cabeçalho de regras e formatação.

Para a linguagem de programação C, a constante MQRFH\_STRUC\_ID\_ARRAY também é definida. Ele tem o mesmo valor que MQRFH\_STRUC\_ID, mas é uma matriz de caracteres em vez de uma cadeia.

### **Versão (MQRFH) para MQRFH**

Este é o número da versão da estrutura; o valor deve ser:

#### **MQRFH\_VERSION\_1**

Version-1 regras e estrutura do cabeçalho de formatação.

O valor inicial desse campo é MQRFH\_VERSION\_1.

### **StrucLength (MQRFH) para MQRFH**

Esse é o comprimento em bytes da estrutura MQRFH, incluindo o campo *NameValueString* no final da estrutura. O comprimento não inclui nenhum dado do usuário que siga o campo *NameValueString*.

Para evitar problemas ao converter os dados do usuário em alguns ambientes, *StrucLength* deve ser um múltiplo de quatro.

A constante a seguir fornece o comprimento da parte *fixa* da estrutura, ou seja, o comprimento excluindo o campo *NameValueString*:

#### **MQRFH\_STRUC\_LENGTH\_FIXED**

Comprimento da parte fixa da estrutura do MQRFH

O valor inicial desse campo é MQRFH\_STRUC\_LENGTH\_FIXED.

### **Codificação (MQLONG) para MQRFH**

Isso especifica a codificação numérica dos dados que seguem *NameValueString*; não se aplica a dados numéricos na própria estrutura MQRFH.

Na chamada MQPUT ou MQPUT1, o aplicativo deve configurar esse campo para o valor apropriado aos dados.

O valor inicial desse campo é MQENC\_NATIVE.

### **CodedCharSetId (MQLONG) para MQRFH**

Isso especifica o identificador do conjunto de caracteres dos dados que seguem *NameValueString*; ele não se aplica aos dados de caracteres na própria estrutura MQRFH

Na chamada MQPUT ou MQPUT1, o aplicativo deve configurar esse campo para o valor apropriado aos dados. O seguinte valor especial pode ser usado:

#### **MQCCSI\_INHERIT**

Os dados de caractere nos dados *após* essa estrutura estão no mesmo conjunto de caracteres que essa estrutura

O gerenciador de filas altera esse valor na estrutura enviada na mensagem para o identificador do conjunto de caractere real da estrutura.. Desde que não ocorra nenhum erro, o valor MQCCSI\_INHERIT não será retornado pela chamada MQGET

MQCCSI\_INHERIT não poderá ser usado se o valor do campo *PutApplType* no MQMD for MQAT\_BROKER..

O valor inicial desse campo é MQCCSI\_UNDEFINED.

### **Formato (MQCHAR8) para MQRFH**

Isso especifica o nome do formato dos dados que seguem *NameValueString*

Na chamada MQPUT ou MQPUT1, o aplicativo deve configurar esse campo para o valor apropriado aos dados. As regras para codificação desse campo são as mesmas que aquelas para o campo *Format* no MQMD.

O valor inicial desse campo é MQFMT\_NONE.

### **Sinalizadores (MQLONG) para MQRFH**

O seguinte pode ser especificado:

#### **MQRFH\_NONE**

Sem bandeiras.

O valor inicial desse campo é MQFRH\_NONE.

### **Sequência NameValue(MQCHARn) para MQRFH**

Esta é uma sequência de caracteres de comprimento variável contendo pares nome-valor no formato:

```
name1 value1 name2 value2 name3 value3 ...
```

Cada nome ou valor deve ser separado do nome ou valor adjacente por um ou mais caracteres em branco; esses espaços em branco não são significativos. Um nome ou valor pode conter espaços em branco significativos prefixando e sufixando o nome ou valor com aspas duplas; todos os caracteres entre as

aspas duplas abertas e as aspas duplas de fechamento correspondentes são tratados como significativos. No exemplo a seguir, o nome é FAMOUS\_WORDS e o valor é Hello World:

```
FAMOUS_WORDS "Hello World"
```

Um nome ou valor pode conter quaisquer caracteres diferentes do caractere nulo (que age como um delimitador para *NameValueString*). No entanto, para auxiliar na interoperabilidade, um aplicativo pode restringir nomes aos seguintes caracteres:

- Primeiro caractere: alfabético maiúsculo ou minúsculo (A a Z ou a a z) ou sublinhado.
- Caracteres subsequentes: alfabético maiúsculo ou minúsculo, dígito decimal (0 a 9), sublinhado, hífen ou ponto.

Se um nome ou valor contiver uma ou mais aspas duplas, o nome ou o valor deverá ser colocado entre aspas duplas e cada aspas duplas dentro da sequência deverá ser duplicado:

```
Famous_Words "The program displayed ""Hello World"""
```

Os nomes e valores fazem distinção entre maiúsculas e minúsculas, ou seja, as letras minúsculas não são consideradas iguais às letras maiúsculas. Por exemplo, FAMOUS\_WORDS e Famous\_Words são dois nomes diferentes.

O comprimento em bytes de *NameValueString* é igual a *StrucLength* menos MQRFH\_STRUC\_LENGTH\_FIXED. Para evitar problemas ao converter os dados do usuário em alguns ambientes, torne esse comprimento um múltiplo de quatro. Preencha *NameValueString* com espaços em branco para esse comprimento ou finalize-o anteriormente, colocando um caractere nulo após o último caractere significativo na cadeia. O caractere nulo e os bytes após ele, até o comprimento especificado de *NameValueString*, são ignorados.

**Nota:** Como o comprimento desse campo não é fixo, o campo é omitido das declarações da estrutura fornecidas para as linguagens de programação suportadas.

## MQRFH2 - Regras e Formatação do Cabeçalho 2

O cabeçalho MQRFH2 é baseado no cabeçalho MQRFH , mas permite que sequências Unicode sejam transportadas sem conversão e pode transportar tipos de dados numéricos. A estrutura MQRFH2 define o formato das regras e do cabeçalho de formatação version-2 . Você usa esse cabeçalho para enviar dados que foram codificados usando uma sintaxe semelhante a XML (Linguagem de Marcação Extensível) Uma mensagem pode conter duas ou mais estruturas MQRFH2 em série, com dados do usuário opcionalmente seguindo a última estrutura MQRFH2 na série.

### Disponibilidade

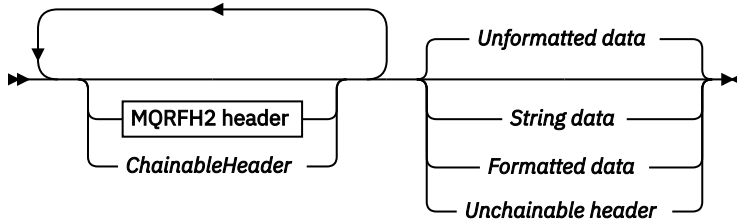
Todos os sistemas IBM MQ , além de IBM MQ MQI clients conectados a esses sistemas

### Nome do Formato

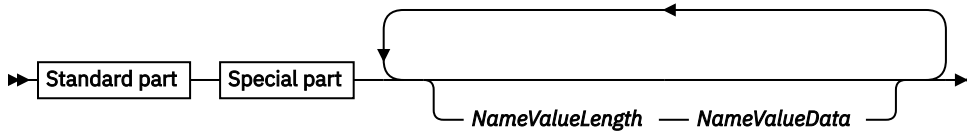
MQFMT\_RF\_HEADER\_2

## Syntax

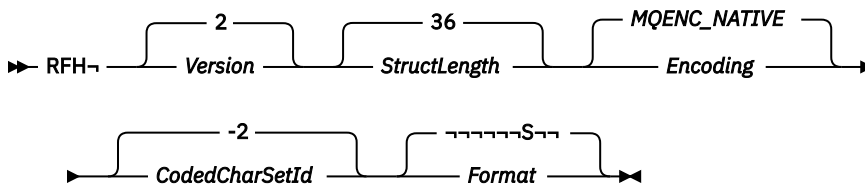
### IBM MQ Message



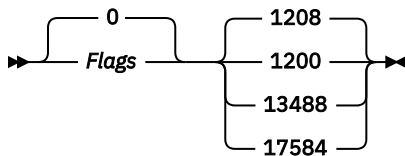
### MQRFH2 header



### Standard part



### Special part



## Conjunto de caracteres e codificação

Regras especiais se aplicam ao conjunto de caracteres e à codificação usados para a estrutura MQRFH2 :

- Campos diferentes de *NameValueData* estão no conjunto de caracteres e na codificação fornecidos pelos campos *CodedCharSetId* e *Encoding* na estrutura do cabeçalho que precede MQRFH2 ou por esses campos na estrutura MQMD se MQRFH2 estiver no início dos dados da mensagem do aplicativo.

O conjunto de caracteres deve ser um que tenha caracteres de byte único para os caracteres válidos em nomes de filas.

Quando MQGMO\_CONVERT é especificado na chamada MQGET , o gerenciador de fila converte os campos MQRFH2 , diferentes de *NameValueData* , para o conjunto de caracteres e codificação solicitados.

- *NameValueData* está no conjunto de caracteres fornecido pelo campo *NameValueCCSID* . Apenas os conjuntos de caracteres Unicode listados são válidos para *NameValueCCSID* ; Consulte a descrição de *NameValueCCSID* para obter detalhes

Alguns conjuntos de caracteres possuem uma representação que depende da codificação.. Se *NameValueCCSID* for um desses conjuntos de caracteres, *NameValueData* deverá estar na mesma codificação que os outros campos no MQRFH2

Quando MQGMO\_CONVERT é especificado na chamada MQGET , o gerenciador de filas converte *NameValueData* para a codificação solicitada, mas não muda seu conjunto de caracteres



## Campos

**Nota:** Na tabela a seguir, os campos são agrupados por uso e não alfabeticamente. Os tópicos filhos seguem a mesma sequência.

<i>Tabela 514. Campos em MQRFH2 para MQRFH2</i>		
<b>Nome de Campo</b>	<b>Nome da constante</b>	<b>Valor da constante</b>
<u>StrucId</u> (identificador de estrutura)	MQRFH_STRUC_ID	'RFH↵'
<u>Versão</u> (número da versão da estrutura).	MQRFH_VERSION_2	2
<u>StrucLength</u> (comprimento em bytes da estrutura MQRFH2)	MQRFH_STRUC_LENGTH_FIXED_2	36
<u>Codificação</u> (codificação numérica dos dados que seguem o último campo <i>NameValueData</i> )	MQENC_NATIVE	Depende do ambiente
<u>CodedCharSetId</u> (identificador do conjunto de caracteres dos dados que seguem o último campo <i>NameValueData</i> )	MQCCSI_INHERIT	-2
<u>Formato</u> (nome do formato dos dados que seguem o último campo <i>NameValueData</i> )	MQFMT_NONE	Espaços em branco
<u>Sinalizadores</u> (sinalizadores)	MQRFH_NONE	0
<u>NameValueCCSID</u> (identificador do conjunto de caracteres codificados dos dados no campo <i>NameValueData</i> )	Nenhum	1208
<u>NameValueLength</u> (comprimento em bytes dos dados no campo <i>NameValueData</i> )	Nenhum	None
<u>NameValueDados</u> (pares nome-valor de propriedades de mensagem)	Nenhum	Nenhum

Tabela 514. Campos em MQRFH2 para MQRFH2 (continuação)

Nome de Campo	Nome da constante	Valor da constante
<b>Notas:</b>		
<p>1. O símbolo ~ representa um único caractere em branco.</p> <p>2. Na linguagem de programação C, a variável macroMQRFH2_DEFAULT contém os valores listados na tabela. Use-o da seguinte maneira para fornecer valores iniciais para os campos na estrutura:</p>		
<pre>MQRFH2 MyRFH2 = {MQRFH2_DEFAULT};</pre>		

## Declarações de idiomas

### Declaração C para MQRFH2

```
typedef struct tagMQRFH2 MQRFH2;
struct tagMQRFH2 {
    MQCHAR4  StrucId;          /* Structure identifier */
    MQLONG   Version;         /* Structure version number */
    MQLONG   StrucLength;     /* Total length of MQRFH2 including all
                             NameValueLength and NameValueData
                             fields */
    MQLONG   Encoding;       /* Numeric encoding of data that follows
                             last NameValueData field */
    MQLONG   CodedCharSetId; /* Character set identifier of data that
                             follows last NameValueData field */
    MQCHAR8  Format;         /* Format name of data that follows last
                             NameValueData field */
    MQLONG   Flags;          /* Flags */
    MQLONG   NameValueCCSID; /* Character set identifier of
                             NameValueData */
};
```

### Declaração COBOL para MQRFH2

```
** MQRFH2 structure
10 MQRFH2.
** Structure identifier
15 MQRFH2-STRUCID PIC X(4).
** Structure version number
15 MQRFH2-VERSION PIC S9(9) BINARY.
** Total length of MQRFH2 including all NAMEVALUELENGTH and
** NAMEVALUEDATA fields
15 MQRFH2-STRUCLength PIC S9(9) BINARY.
** Numeric encoding of data that follows last NAMEVALUEDATA field
15 MQRFH2-ENCODING PIC S9(9) BINARY.
** Character set identifier of data that follows last NAMEVALUEDATA
** field
15 MQRFH2-CODEDCHARSETID PIC S9(9) BINARY.
** Format name of data that follows last NAMEVALUEDATA field
15 MQRFH2-FORMAT PIC X(8).
** Flags
15 MQRFH2-FLAGS PIC S9(9) BINARY.
** Character set identifier of NAMEVALUEDATA
15 MQRFH2-NAMEVALUECCSID PIC S9(9) BINARY.
```

### Declaração de PL/I para MQRFH2

```
dcl
1 MQRFH2 based,
3 StrucId char(4), /* Structure identifier */
3 Version fixed bin(31), /* Structure version number */
3 StrucLength fixed bin(31), /* Total length of MQRFH2 including
                             all NameValueLength and
                             NameValueData fields */
3 Encoding fixed bin(31), /* Numeric encoding of data that
                             follows last NameValueData field */
```

```

3 CodedCharSetId fixed bin(31), /* Character set identifier of data
that follows last NameValueData
field */
3 Format          char(8),      /* Format name of data that follows
last NameValueData field */
3 Flags          fixed bin(31), /* Flags */
3 NameValueCCSID fixed bin(31); /* Character set identifier of
NameValueData */

```

## Declaração do High Level Assembler para MQRFH2

```

MQRFH          DSECT
MQRFH_STRUCID  DS    CL4  Structure identifier
MQRFH_VERSION  DS    F    Structure version number
MQRFH_STRUCLNGTH DS    F    Total length of MQRFH2 including all
* NAMEVALUELENGTH and NAMEVALUEDATA fields
MQRFH_ENCODING DS    F    Numeric encoding of data that follows
* last NAMEVALUEDATA field
MQRFH_CODEDCHARSETID DS    F    Character set identifier of data that
* follows last NAMEVALUEDATA field
MQRFH_FORMAT   DS    CL8  Format name of data that follows last
* NAMEVALUEDATA field
MQRFH_FLAGS    DS    F    Flags
MQRFH_NAMEVALUECCSID DS    F    Character set identifier of
* NAMEVALUEDATA
*
MQRFH_LENGTH   EQU    *-MQRFH
                ORG    MQRFH
MQRFH_AREA     DS    CL(MQRFH_LENGTH)

```

## Declaração do Visual Basic para MQRFH2

```

Type MQRFH2
  StrucId      As String*4 'Structure identifier'
  Version      As Long     'Structure version number'
  StrucLength  As Long     'Total length of MQRFH2 including all
                           'NameValueLength and NameValueData fields'
  Encoding     As Long     'Numeric encoding of data that follows'
                           'last NameValueData field'
  CodedCharSetId As Long   'Character set identifier of data that'
                           'follows last NameValueData field'
  Format        As String*8 'Format name of data that follows last'
                           'NameValueData field'
  Flags        As Long     'Flags'
  NameValueCCSID As Long   'Character set identifier of NameValueData'
End Type

```

### **StrucId (MQLCHAR4) para MQRFH2**

Este é o identificador de estrutura da estrutura de regras e cabeçalho de formatação dois. Ele é sempre um campo de entrada Seu valor é MQRFH2\_STRUC\_ID.

O valor deve ser:.

### **MQRFH2\_STRUC\_ID**

Identificador para a estrutura do cabeçalho dois de regras e de formatação

Para a linguagem de programação C, a constante MQRFH2\_STRUC\_ID\_ARRAY também é definida. Ele tem o mesmo valor que MQRFH2\_STRUC\_ID, mas é uma matriz de caracteres em vez de uma sequência.

### **Versão (MQLONG) para MQRFH2**

Este é o número da versão da estrutura; o valor deve ser:

### **MQRFH\_VERSION\_2**

Version-2 regras e estrutura do cabeçalho de formatação

O valor inicial desse campo é MQRFH\_VERSION\_2.

### **StrucLength (MQLONG) para MQRFH2**

Este é o comprimento em bytes da estrutura MQRFH2 , incluindo os campos *NameValueLength* e *NameValueData* no final da estrutura. É válido haver vários pares de campos *NameValueLength* e *NameValueData* no final da estrutura, na sequência:

```
length1, data1, length2, data2, ...
```

O *StrucLength* não inclui nenhum dado do usuário que possa seguir o último campo *NameValueData* no final da estrutura

Para evitar problemas com a conversão de dados do usuário em alguns ambientes, *StrucLength* deve ser um múltiplo de quatro.

A constante a seguir fornece o comprimento da parte *fixa* da estrutura, ou seja, o comprimento excluindo os campos *NameValueLength* e *NameValueData* :

#### **MQRFH\_STRUC\_LENGTH\_FIXED\_2**

Comprimento da parte fixa da estrutura MQRFH2 .

O valor inicial desse campo é MQRFH\_STRUC\_LENGTH\_FIXED\_2.

#### **Codificação (MQLONG) para MQRFH2**

Isso especifica a codificação numérica dos dados que seguem o último campo *NameValueData* ; ela não se aplica a dados numéricos na própria estrutura MQRFH2

Na chamada MQPUT ou MQPUT1 , o aplicativo deve configurar esse campo para o valor apropriado aos dados.

O valor inicial desse campo é MQENC\_NATIVE.

#### **CodedCharSetId (MQLONG) para MQRFH2**

Isso especifica o identificador do conjunto de caracteres dos dados que seguem o último campo *NameValueData* ; ele não se aplica a dados de caracteres na própria estrutura MQRFH2 .

Na chamada MQPUT ou MQPUT1 , o aplicativo deve configurar esse campo para o valor apropriado aos dados. O seguinte valor especial pode ser usado:

#### **MQCCSI\_INHERIT**

Os dados de caractere nos dados *após* essa estrutura estão no mesmo conjunto de caracteres que essa estrutura

O gerenciador de filas altera esse valor na estrutura enviada na mensagem para o identificador do conjunto de caractere real da estrutura.. Desde que não ocorra nenhum erro, o valor MQCCSI\_INHERIT não será retornado pela chamada MQGET

MQCCSI\_INHERIT não poderá ser usado se o valor do campo *PutApplType* no MQMD for MQAT\_BROKER..

O valor inicial desse campo é MQCCSI\_INHERIT.

#### **Formato (MQCHAR8) para MQRFH2**

Isso especifica o nome do formato dos dados que seguem o último campo *NameValueData*

Na chamada MQPUT ou MQPUT1 , o aplicativo deve configurar esse campo para o valor apropriado aos dados. As regras para codificação desse campo são as mesmas que aquelas para o campo *Format* no MQMD.

O valor inicial desse campo é MQFMT\_NONE.

#### **Sinalizadores (MQLONG) para MQRFH2**

O valor inicial desse campo é MQRFH\_NONE. MQRFH\_NONE deve ser especificado.

## **MQRFH\_NONE**

Sem bandeiras.

## **MQRFH\_INTERNAL**

O cabeçalho MQRFH2 contém propriedades configuradas internamente..

MQRFH\_INTERNAL é para uso do gerenciador de filas.

Os 16 bits principais, MQRFH\_FLAGS\_RESTRICTED\_MASK, são reservados para sinalizadores dos conjuntos do gerenciador de filas Os sinalizadores que um usuário pode configurar são definidos nos 16 bits inferiores

## **NameValueCCSID (MQLONG) para MQRFH2**

Especifica o identificador do conjunto de caracteres codificados dos dados no campo *NameValueData* . Isso é diferente do conjunto de caracteres das outras sequências na estrutura MQRFH2 e pode ser diferente do conjunto de caracteres dos dados (se houver) que seguem o último campo *NameValueData* no final da estrutura.

*NameValueCCSID* deve ter um dos seguintes valores:

<b>CCSID</b>	<b>Significado</b>
1200	UTF-16, a versão Unicode mais recente suportada
13488	UTF-16, o subconjunto da versão Unicode 2.0
17584	UTF-16, o subconjunto da versão Unicode 3.0 (inclui o símbolo do euro)
1208	UTF-8, a versão Unicode mais recente suportada

Para os conjuntos de caracteres UTF-16 , a codificação (ordem de bytes) do *NameValueData* deve ser igual à codificação dos outros campos na estrutura MQRFH2 .

Caracteres além do Unicode Basic Multilingual Plane (aqueles acima de U + FFFF), representados em UTF-16 por pontos de código substitutos (X'D800'a X'DFFF'), ou quatro bytes em UTF-8, não são suportados.

**Nota:** Se *NameValueCCSID* não tiver um dos valores listados acima e a estrutura MQRFH2 requerer conversão na chamada MQGET, a chamada será concluída com o código de razão MQRC\_SOURCE\_CCSID\_ERROR e a mensagem será retornada sem conversão.

O valor inicial deste campo é 1208..

## **NameValue(MQLONG) para MQRFH2**

O comprimento do campo *NameValueData* correspondente

Isso especifica o comprimento em bytes dos dados no campo *NameValueData* . *NameValueLength* deve ser um múltiplo de quatro.

**Nota:** Os campos *NameValueLength* e *NameValueData* são opcionais, mas se presentes, eles devem ocorrer como um par e ser adjacentes. O par de campos pode ser repetido quantas vezes forem necessárias, por exemplo:

```
length1 data1 length2 data2 length3 data3
```

Como esses campos são opcionais, eles são omitidos das declarações da estrutura fornecidas para as várias linguagens de programação suportadas.

## **NameValueData (MQCHARn) para MQRFH2**

*NameValueData* é um campo de comprimento variável que contém uma pasta contendo pares nome-valor de propriedades de mensagem. Uma pasta é uma cadeia de caracteres de comprimento variável contendo dados codificados usando uma sintaxe XML. O comprimento em bytes da sequência de

caracteres é fornecido pelo campo *NameValueLength* que precede o campo *NameValueData* . O comprimento deve ser um múltiplo de quatro.

Os campos *NameValueLength* e *NameValueData* são opcionais, mas se presentes, eles devem ocorrer como um par e ser adjacentes. O par de campos pode ser repetido quantas vezes forem necessárias, por exemplo:

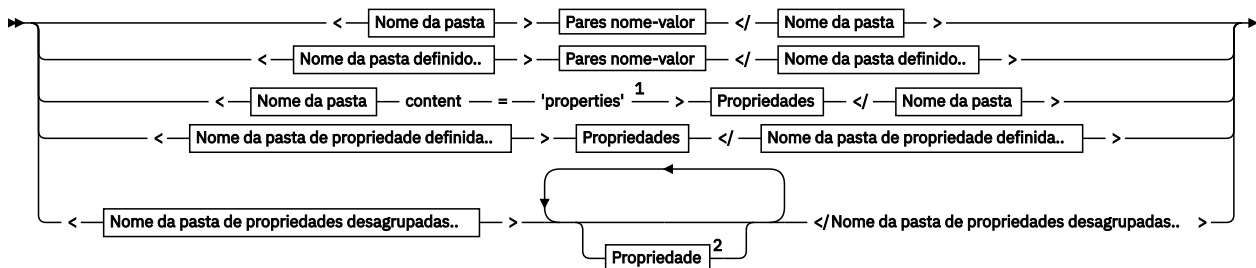
```
length1 data1 length2 data2 length3 data3
```

*NameValueData* não é convertido no conjunto de caracteres especificado na chamada MQGET . Mesmo se a mensagem for recuperada com a opção MQGMO\_CONVERT em vigor *NameValueData* permanece em seu conjunto de caracteres original. No entanto, *NameValueData* é convertido na codificação especificada na chamada MQGET .

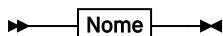
### Notas:

- Como esses campos são opcionais, eles são omitidos das declarações da estrutura fornecidas para as várias linguagens de programação suportadas.
- Os termos "definidos" e "reservados" são usados no diagrama de sintaxe "Definido" significa que o nome é usado por IBM MQ "Reservado" significa que o nome é reservado para uso futuro pelo IBM MQ

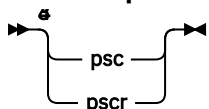
### NameValueData sintaxe



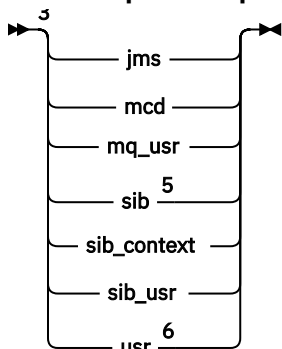
#### Nome da pasta



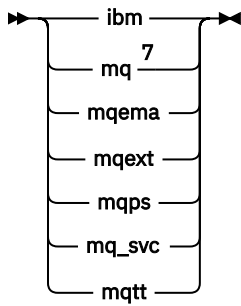
#### Nome da pasta definido..



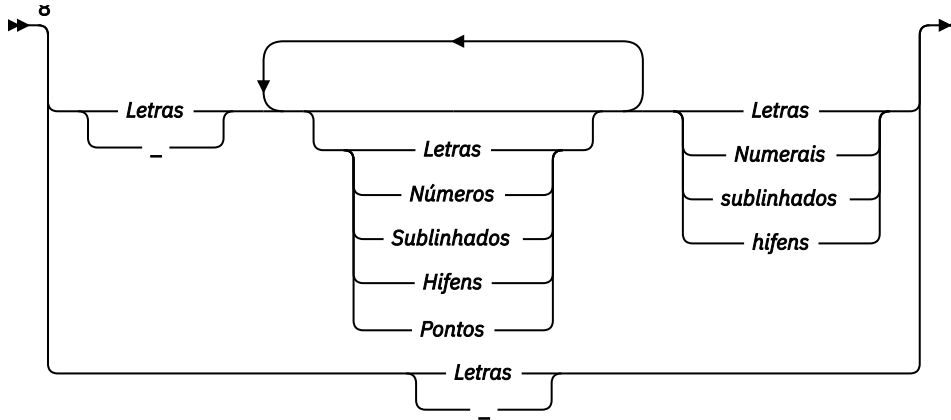
#### Nome da pasta de propriedade definida..



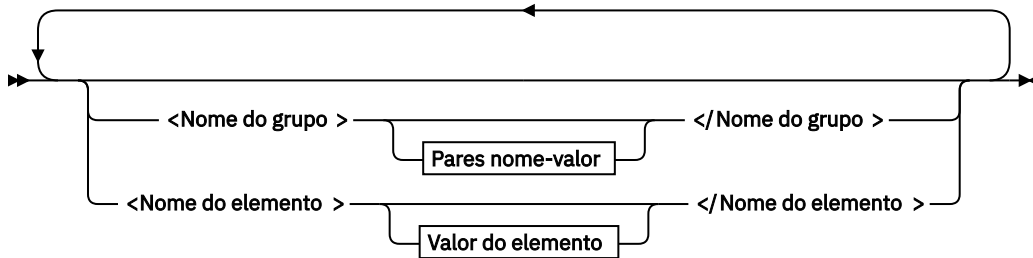
#### Nome da pasta de propriedades desagrupadas..



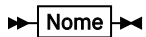
**Nome**



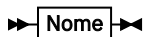
**Pares nome-valor**



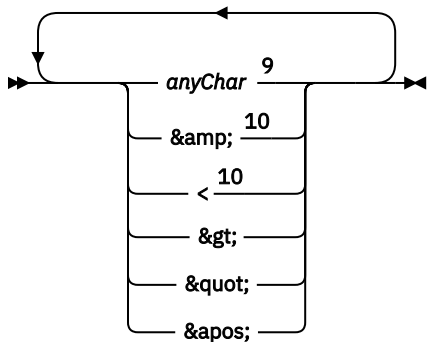
**Nome do grupo**



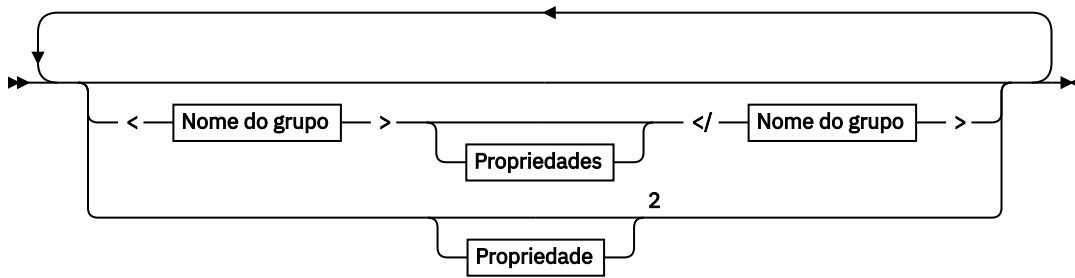
**Nome do elemento**



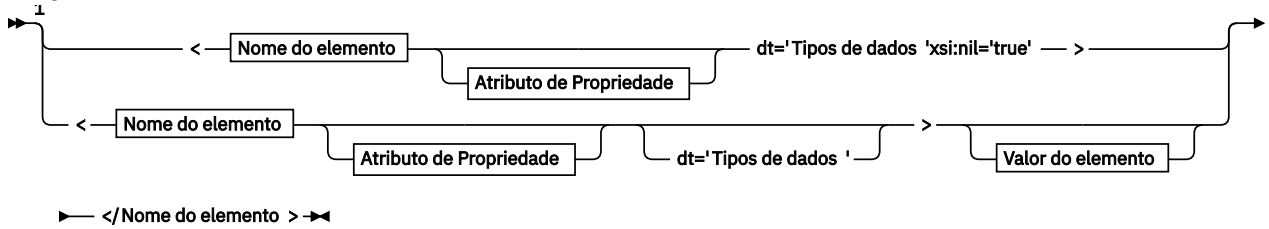
**Valor do elemento**



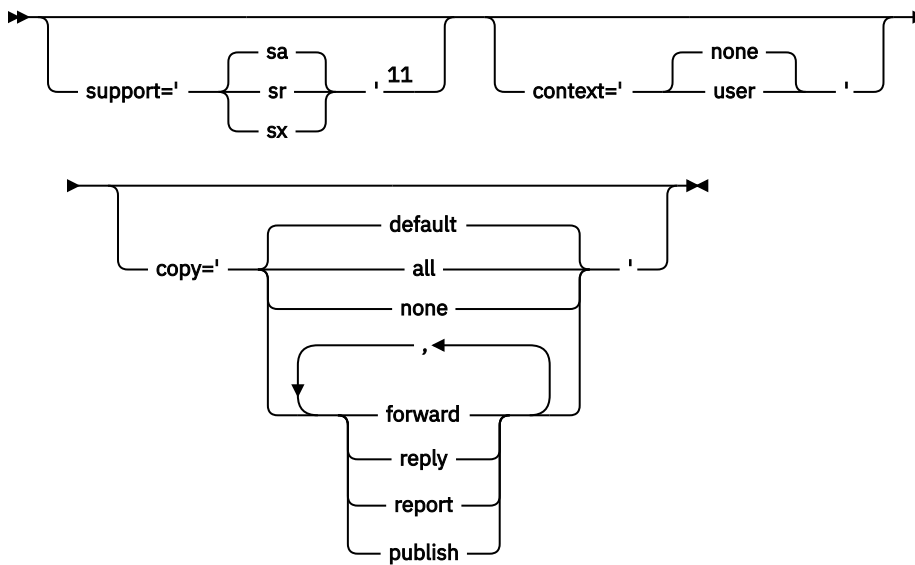
**Propriedades**



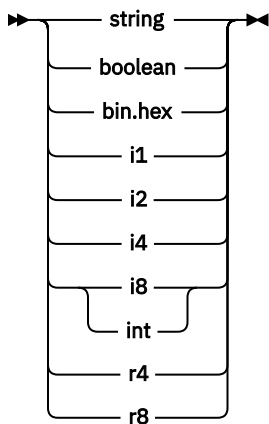
### Propriedade



### Atributo de Propriedade



### Tipos de dados



Notas:

<sup>1</sup> Aspas duplas ou aspas simples são válidas.



- <sup>2</sup> Não use um nome de propriedade inválido; consulte [“Nome de Propriedade Inválido”](#) na página 568 Use um nome de propriedade reservado apenas para seu propósito definido. Consulte [“Nomes de propriedade definidos”](#) na página 568
- <sup>3</sup> O nome deve estar em minúsculas.
- <sup>4</sup> Apenas uma pasta psc e psc:r é suportada
- <sup>5</sup> WebSphere Application Server Serviço Integration Bus ignora sib, sib\_contexte sib\_usr pastas em cabeçalhos MQRFH2 subsequentes e apenas as propriedades no primeiro cabeçalho MQRFH2 são significativas.
- <sup>6</sup> Não mais do que uma pasta usr deve estar presente em um MQRFH2 Propriedades na pasta usr não devem ocorrer mais de uma vez.
- <sup>7</sup> Somente propriedades na primeira pasta mq são significativas. Se a pasta for UTF -8, apenas caracteres de byte único UTF -8 serão suportados. O único caractere de espaço em branco é Unicode U+0020..
- <sup>8</sup> Caracteres válidos são definidos na especificação XML W3C e consistem essencialmente em categorias Unicode Ll, Lu, Lo, Lt, Nl, Mc, Mn, Lm, e Nd ; consulte [Categorias de caracteres Unicode](#).
- <sup>9</sup> Todos os caracteres são significativos. Espaços em branco à esquerda e à direita fazem parte do valor do elemento.
- <sup>10</sup> Não use um caractere inválido; consulte [“Caracteres inválidos”](#) na página 568 Use uma sequência de escape, em vez desses caracteres inválidos.
- <sup>11</sup> O atributo de propriedade de suporte é válido apenas na pasta mq

## Nome da pasta

*NameValueData* contém uma única pasta.. Para criar diversas pastas, crie diversos campos do *NameValueData* É possível criar diversos campos *NameValueData* em um único cabeçalho MQRFH2 dentro de uma mensagem Como alternativa, é possível criar diversos cabeçalhos em cadeia do MQRFH2 , cada um contendo diversos campos *NameValueData*

A ordem dos cabeçalhos MQRFH2 e a ordem dos campos *NameValueData* não fazem diferença para o conteúdo lógico de uma pasta Se a mesma pasta estiver presente mais de uma vez em uma mensagem, ela será analisada como um todo. Se a mesma propriedade ocorrer em várias instâncias da mesma pasta, ela será analisada como uma lista.

Uma análise correta de um MQRFH2 não é afetada pelas maneiras alternativas pelas quais uma pasta pode ser armazenada fisicamente em uma mensagem

Quatro pastas não seguem essa regra Apenas a primeira instância das pastas mq, sib, sib\_contexte sib\_usr são analisadas

Se a mesma propriedade ocorrer mais de uma vez no conteúdo combinado dos cabeçalhos MQRFH2 encadeados, apenas a primeira instância da propriedade será analisada Se uma propriedade for configurada usando uma chamada API, como MQSETMP, e incluída em um MQRFH2 diretamente por um aplicativo, a chamada API terá precedência.

Um nome de pasta é o nome de uma pasta que contém pares ou grupos de nome-valor. Grupos e pares nome-valor podem ser combinados no mesmo nível na árvore de pastas; consulte [Figura 1](#) na página 557. Não combinar um nome de grupo e um nome de elemento; consulte [Figura 2](#) na página 558

```
<group1><nvp1>value</nvp1></group1><group2><nvp2>value</nvp2></group2>
<group3><nvp1>value</nvp1></group3><nvp3>value</nvp3>
```

*Figura 1. Usos corretos de grupos e pares nome-valor*

---

```
<group1><nvp1> value </nvp1> value </group1>
```

*Figura 2. Uso incorreto de grupos e pares nome-valor*

---

Não use um nome de pasta inválido ou reservado; consulte “Nome de caminho inválido” na página 567 e “Nome da pasta reservada ou pasta de propriedade” na página 567 Utilize um nome de pasta definido apenas para seu propósito definido; consulte “Nome da pasta definido..” na página 559

Se você incluir o atributo ' content=properties ' na tag do nome da pasta, a pasta se tornará uma pasta de propriedades; consulte [Figura 3 na página 558](#)

---

```
<myFolder></myfolder>  
<myPropertyFolder contents='properties'></myPropertyFolder>
```

*Figura 3. Exemplo de uma pasta e uma pasta de propriedades*

---

Os nomes de pasta fazem distinção entre letras maiúsculas e minúsculas Nomes de pastas e nomes de pastas de propriedades compartilham o mesmo namespace Eles devem ter nomes diferentes. Folder1 in [Figura 4 na página 558](#) deve ser um nome diferente de Folder2 in [Figura 5 na página 558](#)

---

```
< Folder1 ><NVP1> value </NVP1></ Folder1 >
```

*Figura 4. Espaço de nomes Folder1*

---

```
< Folder2 content='properties'>< Property1 > value </ Property1 ></ Folder2 >
```

*Figura 5. Espaço de nomes Folder2*

---

Grupos, propriedades e pares nome-valor em pastas diferentes têm namespaces diferentes. Property1 in [Figura 5 na página 558](#) é uma propriedade diferente de Property1 in [Figura 6 na página 558](#)

---

```
<Folder3 content='properties'>< Property1 > value </ Property1 ></Folder3>
```

*Figura 6. Espaço de nomes Folder3*

---

As pastas de propriedades são diferentes para pastas não de propriedades em dois aspectos importantes:

1. As pastas de propriedades contêm propriedades e as pastas não propriedades contêm pares nome-valor. As pastas diferem um pouco, sintaticamente.
2. Use as interfaces definidas, como as propriedades MQI ou propriedades de mensagem JMS , para acessar as propriedades de mensagem. As interfaces asseguram que as pastas de propriedade no MQRFH2 estejam bem formadas Uma pasta de propriedades bem formada é interoperável entre gerenciadores de filas em diferentes plataformas e diferentes liberações..

A propriedade MQI é uma maneira robusta de ler e gravar um MQRFH2e evita as dificuldades de analisar um MQRFH2 corretamente.

## Nome da pasta definido..

Um nome de pasta definido é o nome de uma pasta que está reservada para uso pelo IBM MQ ou por outro produto. Não crie uma pasta do mesmo nome e não inclua seus próprios pares nome-valor nas pastas. As pastas definidas são `psc` e `pscr`.

`psc` e `pscr` são usados pela publicação / assinatura enfileiradas

Uma mensagem segmentada colocada com `MQMF_SEGMENT` ou `MQMF_SEGMENTATION_ALLOWED` não pode conter um `MQRFH2` com um nome de pasta definido. O `MQPUT` falha com código de razão 2443, `MQRC_SEGMENTATION_NOT_ALLOWED`.

## Nome da pasta de propriedade definida..

Um nome de pasta de propriedade definido é o nome de uma pasta de propriedade que é usada pelo IBM MQ, ou outro produto. Para obter os nomes das pastas e seus conteúdos, consulte [Pastas de propriedades](#). Os nomes de pasta de propriedade definidos são um subconjunto de todos os nomes de pasta reservados pelo IBM MQ; consulte [“Nome da pasta reservada ou pasta de propriedade”](#) na página 567.

Qualquer elemento armazenado em uma pasta de propriedades definida é uma propriedade. Um elemento armazenado em uma pasta de propriedade definida não deve ter um atributo `content='properties'`.

É possível incluir propriedades apenas nas pastas de propriedade definidas `usr`, `mq_usr`, `sib_usr`. Em outras pastas de propriedade, como `mq` e `sib`, IBM MQ ignora ou lança propriedades que não reconhece.

A descrição de cada pasta de propriedade definida lista as propriedades que o IBM MQ definiu que podem ser usadas por programas de aplicativo. Algumas das propriedades são acessadas indiretamente pela configuração ou obtenção de uma propriedade JMS e algumas são acessadas diretamente usando as chamadas `MQI`, `MQSETMP` e `MQINQMP`.

As pastas de propriedades definidas também contêm outras propriedades que o IBM MQ reservou, mas às quais os aplicativos não têm acesso. Os nomes das propriedades reservadas não são listados. Nenhuma propriedade reservada está presente nas pastas de propriedades `usr`, `mq_usr`, `sib_usr`. Mas não crie propriedades com nomes de propriedades inválidos; consulte [“Nome de Propriedade Inválido”](#) na página 568.

## Pastas de propriedades..

### jms

O `jms` contém campos de cabeçalho JMS, e propriedades JMSX que não são totalmente expressas no `MQMD`. A pasta `jms` está sempre presente em um `JMS MQRFH2`.

Sinônimo da Propriedade	Nome da Propriedade	Tipo de Dados	Pasta
JMSDestination	<code>jms.Dst</code>	string	<code>&lt;jms&gt;&lt;Dst&gt; destination &lt;/Dst&gt;&lt;/jms&gt;</code>
JMSExpiration	<code>jms.Exp</code>	i8	<code>&lt;jms&gt;&lt;Exp&gt; expiration &lt;/Exp&gt;&lt;/jms&gt;</code>
JMSCorrelação	<code>jms.Cid</code>	string	<code>&lt;jms&gt;&lt;Cid&gt; correlationId &lt;/Cid&gt;&lt;/jms&gt;</code>
JMSDelivery	<code>jms.Dlv</code>	i4	<code>&lt;jms&gt;&lt;Dlv&gt; delivery &lt;/Dlv&gt;&lt;/jms&gt;</code>

<i>Tabela 515. nome da propriedade jms, sinônimo, tipo de dados e pasta (continuação)</i>			
<b>Sinônimo da Propriedade</b>	<b>Nome da Propriedade</b>	<b>Tipo de Dados</b>	<b>Pasta</b>
JMSPriority	jms.Pri	i4	<jms><Pri> <i>priority</i> </Pri></jms>
JMSReplyTo	jms.Rto	string	<jms><Rto> <i>replyToURI</i> </Rto></jms>
JMSTimestamp	jms.Tms	i8	<jms><Tms> <i>timestamp</i> </Tms></jms>
JMSXGroupID	jms.Gid	string	<jms><Gid> <i>groupId</i> </Gid></jms>
JMSXGroupSeq	jms.Seq	i4	<jms><Seq> <i>messageSequenceNo</i> </Seq></jms>

Não inclua suas próprias propriedades na pasta jms.

### **mcd**

O mcd contém propriedades que descrevem o formato da mensagem. Por exemplo, a propriedade Msd do domínio de serviço de mensagem identifica uma mensagem JMS como sendo JMSTextMessage, JMSBytesMessage, JMSStreamMessage, JMSMapMessage, JMSObjectMessage ou nula.

A pasta mcd está sempre presente em uma mensagem JMS que contém um MQRFH2.

Ela está sempre presente em uma mensagem que contém um MQRFH2 enviado de IBM Integration Bus. Isso descreve o domínio, o formato, o tipo e o conjunto de mensagens de uma mensagem.

<i>Tabela 516. mcd nome da propriedade, sinônimo, tipo de dados e pasta</i>			
<b>Sinônimo da Propriedade</b>	<b>Nome da Propriedade</b>	<b>Tipo de Dados</b>	<b>Pasta</b>
	mcd.Msd	string	<mcd><Msd> <i>messageDomain</i> </Msd></mcd>
	mcd.Set	string	<mcd><Set> <i>messageDomain</i> </Set></mcd>
	mcd.Type	string	<mcd><Type> <i>messageDomain</i> </Type></mcd>
	mcd.Fmt	string	<mcd><Fmt> <i>messageDomain</i> </Fmt></mcd>

Não inclua suas próprias propriedades na pasta mcd.

### **mq\_usr**

mq\_usr contém propriedades definidas pelo aplicativo que não são expostas como propriedades definidas pelo usuário JMS . As propriedades que não atendem aos requisitos do JMS podem ser colocadas nesta pasta

É possível criar propriedades na pasta mq\_usr .. As propriedades criadas no mq\_usr são semelhantes às propriedades criadas em novas pastas com o atributo content= 'properties' .

## sib

sib contém WebSphere Application Server propriedades de mensagem do sistema do barramento de integração de serviços (WAS/SIB). As propriedades sib não são expostas como propriedades JMS para aplicativos IBM MQ JMS porque não são dos tipos suportados. Por exemplo, algumas propriedades sib não podem ser expostas como propriedades JMS porque são matrizes de bytes. Algumas propriedades sib são expostas aos aplicativos WAS/SIB como propriedades JMS\_IBM\_\* ; elas incluem as propriedades de caminhos de roteamento de encaminhamento e reverso

Não inclua suas próprias propriedades na pasta sib.

## sib\_context

sib\_context contém propriedades de mensagem do sistema WAS/SIB que não são expostas aos aplicativos de usuário WAS/SIB ou como propriedades JMS O sib\_context contém propriedades de segurança e transacionais que são usadas para serviços da Web

Não inclua suas próprias propriedades na pasta sib\_context.

## sib\_usr

sib\_usr contém propriedades de mensagem do usuário WAS/SIB que não são expostas como propriedades do usuário JMS porque elas não são de tipos suportados. sib\_usr é exposto a aplicativos WAS/SIB na interface SIMessage ; consulte [Desenvolvendo Integração de Serviço](#).

O tipo de uma propriedade sib\_usr deve ser bin.hex e o valor deve estar no formato correto. Se um aplicativo IBM MQ gravar um elemento digitado bin.hex para a pasta no formato errado, o aplicativo receberá um IOException Se o tipo de dados da propriedade não for bin.hex , o aplicativo receberá um ClassCastException

Não tente tornar as propriedades de usuário do JMS disponíveis para WAS/SIB usando essa pasta; em vez disso, use a pasta usr

É possível criar propriedades na pasta sib\_usr ..

## usr

O usr contém propriedades JMS definidas por aplicativo associadas à mensagem. A pasta usr estará presente apenas quando um aplicativo tiver configurado uma propriedade definida pelo aplicativo.

usr é a pasta de propriedades padrão Se uma propriedade for configurada sem um nome de pasta, ela será colocada na pasta usr

Sinônimo da Propriedade	Nome da Propriedade	Tipo de Dados	Pasta
	usr.contentType	string	<usr><contentType>text/xml; charset=utf-8</contentType></usr>
	usr.endpointURL	string	<usr><endpointURL> URI </endpointURL></usr>
	usr.targetService	string	<usr><targetService> serviceName </targetService></usr>
	usr.soapAction	string	<usr><soapAction> name </soapAction></usr>
	usr.transportVersion	string	<usr><transportVersion> version </transportVersion></usr>

É possível criar propriedades na pasta `usr` ..

Uma mensagem segmentada colocada com `MQMF_SEGMENT` ou `MQMF_SEGMENTATION_ALLOWED` não pode conter um `MQRFH2` com um nome de pasta de propriedades definido. O `MQPUT` falha com código de razão 2443, `MQRC_SEGMENTATION_NOT_ALLOWED`.

## Nome da pasta de propriedades desagrupadas..

### **ibm**

`ibm` contém propriedades que são usadas apenas por IBM MQ..

*Tabela 518. ibm nome da propriedade, sinônimo, tipo de dados e pasta*

<b>Sinônimo da Propriedade</b>	<b>Nome da Propriedade</b>	<b>Tipo de Dados</b>	<b>Pasta</b>
	<code>ibm.rfp</code>	<code>string</code>	<code>&lt;ibm&gt;&lt;rfp&gt;fingerprint&lt;/rfp&gt;&lt;/ibm&gt;</code>

Não inclua suas próprias propriedades na pasta `ibm`.

### **mq**

`mq` contém propriedades que são usadas apenas por IBM MQ..

As seguintes restrições se aplicam às propriedades na pasta `mq` :

- Apenas as propriedades na primeira pasta `mq` significativa na mensagem são atuadas pelo MQ; as propriedades em qualquer outra pasta `mq` na mensagem são ignoradas.
- Apenas caracteres UTF-8 de byte único são permitidos na pasta. Um caractere de multibyte na pasta pode causar falha na análise e a mensagem ser rejeitada.
- Não use sequências de escape na pasta. Uma sequência de escape é tratada como o valor real do elemento.
- Apenas o caractere Unicode U+0020 é tratado como espaço em branco na pasta. Todos os outros caracteres são tratados como significativos e podem causar falha na análise da pasta e a mensagem ser rejeitada.

Se a análise da pasta `mq` falhar ou a pasta não observar essas restrições, a mensagem será rejeitada com o código de razão 2527, `MQRC_RFH_RESTRICTED_FORMAT_ERR`.

Não inclua suas próprias propriedades na pasta `mq`.

### **mqema**

`mqema` contém propriedades que são usadas apenas por WebSphere Application Server.. A pasta foi substituída por `mqext`..

Não inclua suas próprias propriedades na pasta `mqema`.

### **mqext**

O `mqext` contém os seguintes tipos de propriedade:

- Propriedades que são usadas somente pelo WebSphere Application Server.
- Propriedades relacionadas ao atraso na entrega de mensagens.

A pasta somente estará presente se o aplicativo tiver configurado pelo menos uma das propriedades definidas do IBM ou usado o atraso de entrega.

<i>Tabela 519. mqext nome da propriedade, sinônimo, tipo de dados e pasta</i>			
<b>Sinônimo da Propriedade</b>	<b>Nome da Propriedade</b>	<b>Tipo de Dados</b>	<b>Pasta</b>
JMSArmCorrelator	mqext.Arm	string	<mqext><Arm>armCorrelator</Arm></mqext>
JMSRMCorrelator	mqext.Wrm	string	<mqext><Wrm>wrmCorrelator</Wrm></mqext>
JMSDeliveryTime	mqext.Dlt	i8	<mqext><Dlt>DeliveryTime</Dlt></mqext>
JMSDeliveryDelay	mqext.Dly	i8	<mqext><Dly>DeliveryTime</Dly></mqext>

Não inclua suas próprias propriedades na pasta mqext.

### **mqps**

O mqps contém propriedades usadas apenas por IBM MQ publicar/assinar. A pasta estará presente somente se o aplicativo tiver configurado pelo menos uma das propriedades de publicação/assinatura integradas.

<i>Tabela 520. mqps nome da propriedade, sinônimo, tipo de dados e pasta</i>			
<b>Sinônimo da Propriedade</b>	<b>Nome da Propriedade</b>	<b>Tipo de Dados</b>	<b>Pasta</b>
MQTopicString	mqps.Top	string	<mqps><Top>topicString</Top></mqps>
MQSubUserData	mqps.Sud	string	<mqps><Sud>subscriberUserData...</Sud></mqps>
MQIsRetained	mqps.Ret	boolean	<mqps><Ret>isRetained</Ret></mqps>
MQPubOptions	mqps.Pub	i8	<mqps><Pub>publicationOptions</Pub></mqps>
MQPubLevel	mqps.Pbl	i8	<mqps><Pbl>publicationLevel</Pbl></mqps>
MQPubTime	mqpse.Pts	string	<mqps><Pts>publicationTime</Pts></mqps>
MQPubSeqNum	mqpse.Seq	i8	<mqps><Seq>publicationSequenceNumber</Seq></mqps>
MQPubStrInpData	mqpse.Sid	string	<mqps><Sid>publicationData</Sid></mqps>
MQPubFormat	mqpse.Pfmt	i8	<mqps><Pfmt>messageFormat</Pfmt></mqps>

Não inclua suas próprias propriedades na pasta mqps.

### **mq\_svc**

mq\_svc contém as propriedades usadas por SupportPac MA93

Não inclua suas próprias propriedades na pasta mq\_svc.

### **mqtt**

mqtt contém propriedades usadas por MQ Telemetry

Tabela 521. nome da propriedade mqtt, sinônimo, tipo de dados e pasta

Sinônimo da Propriedade	Nome da Propriedade	Tipo de Dados	Pasta
	mqtt.clientId	string	<mqtt><clientId> <i>topicString</i> </clientId></mqtt>
	mqtt.qos	i4	<mqtt><qos> <i>qualityOfService</i> </qos></mqtt>
	mqtt.msgid	string	<mqtt><msgid> <i>messageIdentifier</i> </msgid></mqtt>

Não inclua suas próprias propriedades na pasta mqtt.

Uma mensagem segmentada colocada com MQMF\_SEGMENT ou MQMF\_SEGMENTATION\_ALLOWED não pode conter um MQRFH2 com um nome de pasta de propriedades desagrupadas O MQPUT falha com código de razão 2443, MQRC\_SEGMENTATION\_NOT\_ALLOWED.

## Pares nome-valor

No diagrama de sintaxe, "pares nome-valor" descreve o conteúdo de uma pasta comum. Uma pasta comum contém grupos e elementos. Um elemento é um par nome-valor. Um grupo contém elementos e outros grupos..

Em termos de árvores, os elementos são nós folha e os grupos são nós internos. Um nó interno e a pasta, que é o nó raiz, podem conter uma mistura de nós internos e nós folha. Um nó não pode ser um nó interno e um nó folha ao mesmo tempo; consulte [Figura 2 na página 558](#).

## Propriedades

No diagrama de sintaxe, "Propriedades" descreve o conteúdo de uma pasta de propriedades.. Uma pasta de propriedades contém grupos, e propriedades. Uma propriedade é um par de nome-valor com um atributo de tipo de dados opcional. Um grupo contém propriedades e outros grupos..

Em termos de árvores, propriedades são nós folhas e grupos são nós internos. Um nó interno, e a pasta de propriedades, que é o nó raiz, podem conter uma mistura de nós internos e nós folha.. Um nó não pode ser um nó interno e um nó folha ao mesmo tempo; consulte [Figura 2 na página 558](#).

## Propriedade

Uma propriedade de mensagem é um par nome-valor em uma pasta de propriedades.. Ele pode, opcionalmente, incluir um atributo de tipo de dados e um atributo de propriedade; por exemplo, consulte o código a seguir. Se o atributo de tipo de dados for omitido, o tipo de propriedade será `string`

```
<pf><p1 dt='i8' > value </p1></pf>
```

O nome de uma propriedade de mensagem é seu nome do caminho completo, com a sintaxe XML-like, <>, substituída por pontos Por exemplo, `myPropertyFolder1.myGroup1.myGroup2.myProperty1` é mapeado para uma sequência `NameValueData`, conforme a seguir: A sequência é formatado para facilitar a leitura.

```
<myPropertyFolder1>
  <myGroup1>
    <myGroup2>
      <myProperty1>value</myProperty1>
    </myGroup2>
  </myGroup1>
</myPropertyFolder1>
```

Uma pasta de propriedades pode conter várias propriedades Por exemplo, as propriedades em [Figura 7 na página 565](#) são mapeados para a pasta de propriedades em [Mudanças](#) para isolar a fila de vendas em um novo cluster e separar as filas de transmissão do cluster de gateway.



```
myPropertyFolder1.myProperty4
myPropertyFolder1.myGroup1.myGroup2.myProperty1
myPropertyFolder1.myGroup1.myGroup2.myProperty2
myPropertyFolder1.myGroup1.myProperty3
```

Figura 7. Várias propriedades com o mesmo nome raiz

```
<myPropertyFolder1>
  <myProperty4>value</myProperty4>
  <myGroup1>
    <myGroup2>
      <myProperty1>value</myProperty1>
      <myProperty2>value</myProperty2>
    </myGroup2>
    <myProperty3>value</myProperty3>
  </myGroup1>
</myPropertyFolder1>
```

Figura 8. Mapeamento de vários nomes de propriedades

## Nome

Um nome deve começar com uma *Carta* ou um *sublinhado*. Ele não deve conter um *Colon*, não terminar em um *Período* e conter apenas *Letras*, *Numéricos*, *sublinhados*, *hifense pontos*. Caracteres válidos são definidos na especificação XML W3C e consistem essencialmente em categorias Unicode L1, Lu, Lo, Lt, Nl, Mc, Mn, Lm, e Nd; consulte [Categorias de caracteres Unicode](#).

O caminho completo de uma propriedade ou par nome-valor não deve quebrar a regra descrita em “Nome de caminho inválido” na página 567. Os caminhos são restritos a 4095 bytes, não devem conter caracteres de compatibilidade Unicode e não devem iniciar com a cadeia XML

## Nome do grupo

Um nome de grupo tem a mesma sintaxe que um nome.. Os nomes de grupos são opcionais Propriedades e pares nome-valor podem ser colocados na raiz de uma pasta. Use grupos se ajudar a organizar propriedades e pares nome-valor.

## Nome do elemento

Um nome de elemento tem a mesma sintaxe que um nome.

## Valor do elemento

Um valor de elemento inclui todo o espaço em branco entre a tag `< Element name >` e `< /Element name >`. Não use os dois caracteres `<` e `&` em um valor. Substitua então por `< e & ;`.

## Atributos de propriedade

Os atributos de propriedade mapeiam campos do descritor de propriedade: Os mapeamentos são os seguintes:

### Suporte

#### sa (padrão)

MQPD\_SUPPORT\_OPTIONAL

#### solicitação de serviço

MQPD\_SUPPORT\_REQUIRED

**sx**  
MQPD\_SUPPORT\_REQUIRED\_IF\_LOCAL

### Context

**none (padrão)**  
MQPD\_NO\_CONTEXT

**user**  
MQPD\_USER\_CONTEXT

### CopyOptions

**encaminhamento**  
MQPD\_COPY\_FORWARD

**resposta**  
MQPD\_COPY\_REPLY

**relatório**  
MQPD\_COPY\_REPORT

**publicar**  
MQPD\_COPY\_PUBLISH

**all**  
MQPD\_COPY\_ALL

Não use all em combinação com outras opções

**padrão**  
MQPD\_COPY\_DEFAULT

Não use default em combinação com outras opções.. default é igual a forward + report + publish.

**Nenhum**  
MQPD\_COPY\_NONE

Não use none em combinação com outras opções

Os atributos de propriedade de Suporte são aplicáveis apenas às propriedades na pasta mq

Os atributos de propriedade Contexto e CopyOptions são aplicáveis a todas as pastas de propriedade

## Tipos de dados

Os tipos de dados MQRFH2 são mapeados para os tipos de propriedade de mensagem da seguinte forma:

<i>Tabela 522. Mapeamentos de Tipos de Dados</i>	
<b>Tipo de dados do MQRFH2</b>	<b>Tipo propriedade mensagem</b>
bin.hex	MQBYTE[]
boolean	MQBOOL
i1	MQINT8
i2	MQINT16
i4	MQINT32
i8	MQINT64
r4	MQFLOAT32
r8	MQFLOAT64
string	MQCHAR[]

Qualquer elemento sem um tipo de dados é considerado do tipo `string`.

Um valor nulo é indicado pelo atributo do elemento `xsi:nil= 'true'` Não use o atributo `xsi:nil= 'false'` para valores não nulos Por exemplo, a propriedade a seguir possui um valor nulo:

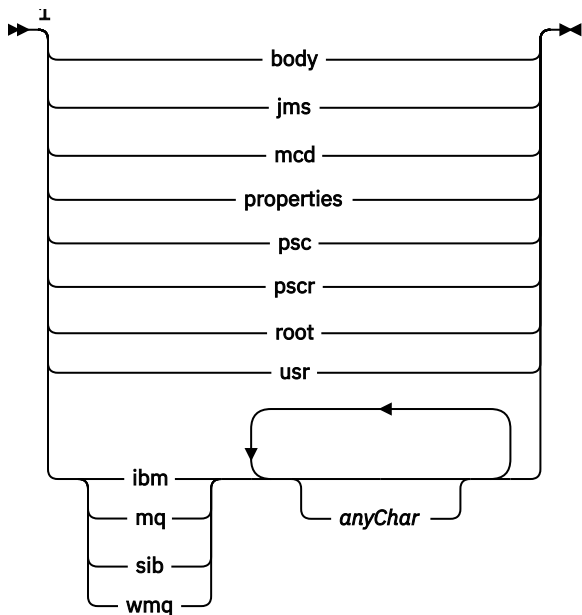
```
<NullProperty  
xsi:nil='true'></NullProperty>
```

Uma propriedade de cadeia de bytes ou caracteres pode ter um valor vazio. Um valor vazio é representado por um elemento `EMPTY` com um valor de elemento de comprimento zero Por exemplo, a propriedade a seguir possui um valor vazio:

```
<EmptyProperty></EmptyProperty>
```

### Nome da pasta reservada ou pasta de propriedade

Restrinja o nome de uma pasta ou pasta de propriedades para não iniciar com nenhuma das sequências a seguir. Os prefixos são reservados para nomes de pasta ou de propriedade criados por IBM

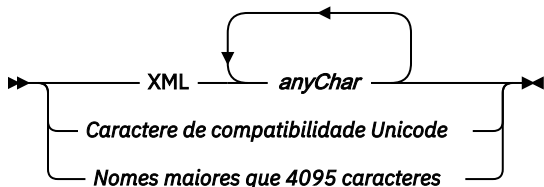


Notas:

<sup>1</sup> Uma pasta reservada ou um nome de propriedade contém qualquer mistura de letras minúsculas e maiúsculas.

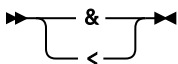
### Nome de caminho inválido

Restrinja o caminho completo de um par nome-valor ou uma propriedade para não incluir nenhuma das seguintes sequências.



## Caracteres inválidos

Sempre use as seqüências de escape `&amp;` e `<` em vez dos literais `"&"` e `"<"`..

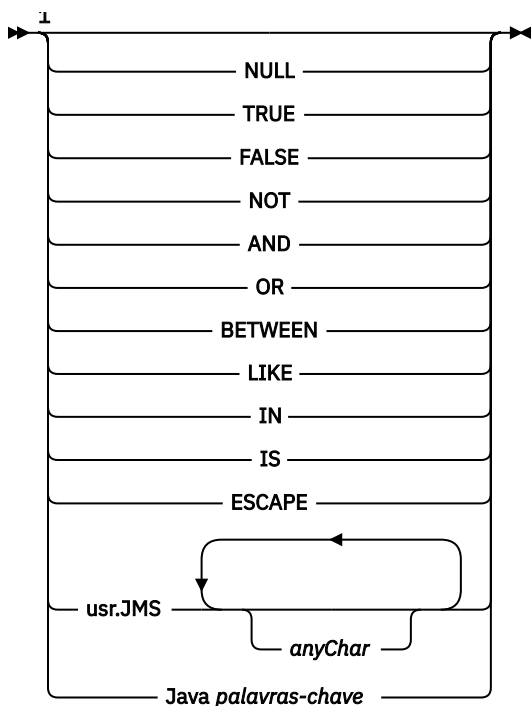


## Nomes de propriedade definidos

Os nomes de propriedades definidos são os nomes de propriedades que são definidos por IBM MQ ou outros produtos e usados por IBM MQ e aplicativos de usuário. As propriedades definidas existem apenas em pastas de propriedades definidas. Os nomes de propriedades definidos são descritos na descrição de pastas de propriedades; consulte [Pastas de propriedades](#).

## Nome de Propriedade Inválido

Não construa nomes de propriedades que correspondam à regra a seguir: A regra se aplica ao caminho de propriedade completo que nomeia uma propriedade e não apenas ao nome do elemento de propriedade.



Notas:

<sup>1</sup> Um nome de propriedade inválido pode conter qualquer combinação de maiúsculas e minúsculas.

## Atributos inválidos

Pastas de propriedades e propriedades podem incluir apenas [“Atributos de propriedade”](#) na página 565 e [“Tipos de dados”](#) na página 566 suportados.

Quaisquer atributos semelhantes a XML não suportados, por exemplo, nomes com valores de seqüência de caracteres entre aspas, que são incluídos em pastas de propriedades ou propriedades podem ser removidos.

Atributos semelhantes a XML incluídos em pastas sem propriedade ou elementos sem propriedade que permanecem em cabeçalhos MQRFH2.

## MQRMH - cabeçalho de mensagem de referência

A estrutura MQRMH define o formato de um cabeçalho da mensagem de referência.. Esse cabeçalho é usado com saídas de canal de mensagens gravadas pelo usuário para enviar quantidades extremamente grandes de dados (chamados *dados em massa* ) de um gerenciador de filas para outro. A diferença em relação ao sistema de mensagens normal é que os dados em massa não são armazenados em uma fila; em vez disso, apenas uma *referência* aos dados em massa é armazenada na fila. Isso reduz a possibilidade de os recursos do IBM MQ serem esgotados por um pequeno número de mensagens extremamente grandes

### Disponibilidade

A estrutura MQRMH está disponível nas plataformas a seguir:

-  AIX
-  IBM i
-  Linux
-  Windows

e para clientes IBM MQ conectados a esses sistemas.

### Nome do Formato

MQFMT\_REF\_MSG\_HEADER

### Conjunto de caracteres e codificação

Os dados de caractere em MQRMH e as sequências endereçadas pelos campos de deslocamento, devem estar no conjunto de caracteres do gerenciador de filas locais; isso é fornecido pelo atributo do gerenciador de filas **CodedCharSetId** Dados numéricos em MQRMH devem estar na codificação da máquina nativa; isso é fornecido pelo valor de MQENC\_NATIVE para a linguagem de programação C.

Configure o conjunto de caracteres e a codificação do MQRMH nos campos *CodedCharSetId* e *Encoding* em:

- O MQMD (se a estrutura MQRMH estiver no início dos dados da mensagem) ou
- A estrutura do cabeçalho que precede a estrutura MQRMH (todos os outros casos).

### Uso

Um aplicativo coloca uma mensagem consistindo em um MQRMH, mas omitindo os dados em massa Quando um agente do canal de mensagens (MCA) lê a mensagem da fila de transmissão, uma saída de mensagem fornecida pelo usuário é chamada para processar o cabeçalho da mensagem de referência. A saída pode anexar à mensagem de referência os dados em massa identificados pela estrutura MQRMH, antes que o MCA envie a mensagem através do canal para o próximo gerenciador de filas.

Na extremidade de recebimento, uma saída de mensagem que aguarda mensagens de referência deve existir.. Quando uma mensagem de referência é recebida, a saída deve criar o objeto dos dados em massa que seguem o MQRMH na mensagem e, em seguida, transmitir a mensagem de referência sem os dados em massa. A mensagem de referência pode ser recuperada posteriormente por um aplicativo que lê a mensagem de referência (sem os dados em massa) a partir de uma fila.

Normalmente, a estrutura MQRMH é tudo o que está na mensagem No entanto, se a mensagem estiver em uma fila de transmissão, um ou mais cabeçalhos adicionais precedem a estrutura MQRMH..

Uma mensagem de referência também pode ser enviada para uma lista de distribuição Neste caso, a estrutura MQDH e seus registros relacionados precedem a estrutura MQRMH quando a mensagem está em uma fila de transmissão

**Nota:** Não envie uma mensagem de referência como uma mensagem segmentado, porque a saída de mensagem não pode processá-la corretamente

## Conversão de Dados

Para propósitos de conversão de dados, a conversão da estrutura MQRMH inclui a conversão dos dados do ambiente de origem, nome do objeto de origem, dados do ambiente de destino e nome do objeto de destino. Quaisquer outros bytes dentro de *StrucLength* bytes do início da estrutura são descartadas ou têm valores indefinidos após a conversão de dados Os dados em massa são convertidos desde que todas as instruções a seguir sejam verdadeiras:

- Os dados em massa estão presentes na mensagem quando a conversão de dados é executada..
- O campo *Format* no MQRMH possui um valor diferente de MQFMT\_NONE.
- Existe uma saída de conversão de dados gravados pelo usuário com o nome do formato especificado.

Esteja ciente, no entanto, de que geralmente os dados em massa não estão presentes na mensagem quando a mensagem está em uma fila e que, como resultado, os dados em massa são convertidos pela opção MQGMO\_CONVERT.

## Campos

**Nota:** Na tabela a seguir, os campos são agrupados por uso e não alfabeticamente. Os tópicos filhos seguem a mesma sequência.

<i>Tabela 523. Campos em MQRMH para MQRMH.</i>		
Nome e descrição do campo	Nome da constante	Valor inicial (se houver) da constante
<u>StrucId</u> (identificador de estrutura)	MQRMH_STRUC_ID	'RMH→'
<u>Versão</u> (número da versão da estrutura).	MQRMH_VERSION_1	1
<u>StrucLength</u> (comprimento total de MQRMH, incluindo sequências no final de campos fixos, mas não os dados em massa)	Nenhum	0
<u>Codificação</u> (codificação numérica de dados em massa)	MQENC_NATIVE	Depende do ambiente
<u>CodedCharSetId</u> (identificador do conjunto de caracteres de dados em massa)..	MQCCSI_UNDEFINED	0
<u>Formato</u> (nome do formato de dados em massa)	MQFMT_NONE	Espaços em branco
<u>Flags</u> (sinalizadores de mensagem de referência)	MQRMHF_NOT_LAST	0
<u>ObjectType</u> (tipo de objeto)	Nenhum	Espaços em branco
<u>ObjectInstanceId</u> (identificador de instância do objeto)	MQOII_NONE	Nulos
<u>SrcEnvComprimento</u> (comprimento dos dados do ambiente de origem)	Nenhum	0
<u>SrcEnvOffset</u> (deslocamento de dados do ambiente de origem)	Nenhum	0
<u>SrcNameComprimento</u> (comprimento do nome do objeto de origem)	Nenhum	0
<u>SrcNameOffset</u> (deslocamento do nome do objeto de origem)	Nenhum	0

Tabela 523. Campos em MQRMH para MQRMH. (continuação)

Nome e descrição do campo	Nome da constante	Valor inicial (se houver) da constante
<u>DestEnvComprimento</u> (comprimento dos dados do ambiente de destino)	Nenhum	0
<u>DestEnvOffset</u> (deslocamento de dados do ambiente de destino)	Nenhum	0
<u>DestNameComprimento</u> (comprimento do nome do objeto de destino)	Nenhum	0
<u>DestNameOffset</u> (deslocamento do nome do objeto de destino)	Nenhum	0
<u>DataLogicalDataLogical</u> (comprimento de dados em massa)	Nenhum	0
<u>DataLogicalOffset</u> (deslocamento baixo de dados em massa)	Nenhum	0
<u>DataLogicalOffset2</u> (deslocamento alto de dados em massa)	Nenhum	0

**Notas:**

1. O símbolo ↵ representa um único caractere em branco.
2. Na linguagem de programação C, a variável macro MQRMH\_DEFAULT contém os valores listados na tabela. Use-o da seguinte maneira para fornecer valores iniciais para os campos na estrutura:

```
MQRMH MyRMH = {MQRMH_DEFAULT};
```

## Declarações de idiomas

### Declaração C para MQRMH

```
typedef struct tagMQRMH MQRMH;
struct tagMQRMH {
    MQCHAR4    StrucId;           /* Structure identifier */
    MQLONG    Version;          /* Structure version number */
    MQLONG    StrucLength;       /* Total length of MQRMH, including
                                strings at end of fixed fields, but
                                not the bulk data */
    MQLONG    Encoding;         /* Numeric encoding of bulk data */
    MQLONG    CodedCharSetId;   /* Character set identifier of bulk
                                data */
    MQCHAR8    Format;          /* Format name of bulk data */
    MQLONG    Flags;           /* Reference message flags */
    MQCHAR8    ObjectType;     /* Object type */
    MQBYTE24   ObjectInstanceId; /* Object instance identifier */
    MQLONG    SrcEnvLength;     /* Length of source environment data */
    MQLONG    SrcEnvOffset;     /* Offset of source environment data */
    MQLONG    SrcNameLength;    /* Length of source object name */
    MQLONG    SrcNameOffset;    /* Offset of source object name */
    MQLONG    DestEnvLength;    /* Length of destination environment
                                data */
    MQLONG    DestEnvOffset;    /* Offset of destination environment
                                data */
    MQLONG    DestNameLength;   /* Length of destination object name */
    MQLONG    DestNameOffset;   /* Offset of destination object name */
    MQLONG    DataLogicalLength; /* Length of bulk data */
    MQLONG    DataLogicalOffset; /* Low offset of bulk data */
    MQLONG    DataLogicalOffset2; /* High offset of bulk data */
};
```

## Declaração COBOL para MQRMH

```
** MQRMH structure
10 MQRMH.
** Structure identifier
15 MQRMH-STRUCID PIC X(4).
** Structure version number
15 MQRMH-VERSION PIC S9(9) BINARY.
** Total length of MQRMH, including strings at end of fixed fields,
** but not the bulk data
15 MQRMH-STRUCLength PIC S9(9) BINARY.
** Numeric encoding of bulk data
15 MQRMH-ENCODING PIC S9(9) BINARY.
** Character set identifier of bulk data
15 MQRMH-CODEDCHARSETID PIC S9(9) BINARY.
** Format name of bulk data
15 MQRMH-FORMAT PIC X(8).
** Reference message flags
15 MQRMH-FLAGS PIC S9(9) BINARY.
** Object type
15 MQRMH-OBJECTTYPE PIC X(8).
** Object instance identifier
15 MQRMH-OBJECTINSTANCEID PIC X(24).
** Length of source environment data
15 MQRMH-SRCENVLENGTH PIC S9(9) BINARY.
** Offset of source environment data
15 MQRMH-SRCENVOFFSET PIC S9(9) BINARY.
** Length of source object name
15 MQRMH-SRCNAMELENGTH PIC S9(9) BINARY.
** Offset of source object name
15 MQRMH-SRCNAMEOFFSET PIC S9(9) BINARY.
** Length of destination environment data
15 MQRMH-DESTENVLENGTH PIC S9(9) BINARY.
** Offset of destination environment data
15 MQRMH-DESTENVOFFSET PIC S9(9) BINARY.
** Length of destination object name
15 MQRMH-DESTNAMELENGTH PIC S9(9) BINARY.
** Offset of destination object name
15 MQRMH-DESTNAMEOFFSET PIC S9(9) BINARY.
** Length of bulk data
15 MQRMH-DATALOGICALLength PIC S9(9) BINARY.
** Low offset of bulk data
15 MQRMH-DATALOGICALOFFSET PIC S9(9) BINARY.
** High offset of bulk data
15 MQRMH-DATALOGICALOFFSET2 PIC S9(9) BINARY.
```

## Declaração PL/I para MQRMH

```
dcl
1 MQRMH based,
3 StrucId char(4), /* Structure identifier */
3 Version fixed bin(31), /* Structure version number */
3 StrucLength fixed bin(31), /* Total length of MQRMH,
including strings at end of
fixed fields, but not the bulk
data */
3 Encoding fixed bin(31), /* Numeric encoding of bulk
data */
3 CodedCharSetId fixed bin(31), /* Character set identifier of
bulk data */
3 Format char(8), /* Format name of bulk data */
3 Flags fixed bin(31), /* Reference message flags */
3 ObjectType char(8), /* Object type */
3 ObjectInstanceId char(24), /* Object instance identifier */
3 SrcEnvLength fixed bin(31), /* Length of source environment
data */
3 SrcEnvOffset fixed bin(31), /* Offset of source environment
data */
3 SrcNameLength fixed bin(31), /* Length of source object name */
3 SrcNameOffset fixed bin(31), /* Offset of source object name */
3 DestEnvLength fixed bin(31), /* Length of destination
environment data */
3 DestEnvOffset fixed bin(31), /* Offset of destination
environment data */
3 DestNameLength fixed bin(31), /* Length of destination object
name */
3 DestNameOffset fixed bin(31), /* Offset of destination object
name */
```



```

3 DataLogicalLength fixed bin(31), /* Length of bulk data */
3 DataLogicalOffset fixed bin(31), /* Low offset of bulk data */
3 DataLogicalOffset2 fixed bin(31); /* High offset of bulk data */

```

### Declaração do High Level Assembler para MQRMH

```

MQRMH          DSECT
MQRMH_STRUCID  DS    CL4   Structure identifier
MQRMH_VERSION  DS    F     Structure version number
MQRMH_STRUCLNGTH DS    F     Total length of MQRMH, including
*              strings at end of fixed fields, but
*              not the bulk data
MQRMH_ENCODING DS    F     Numeric encoding of bulk data
MQRMH_CODEDCHARSETID DS    F   Character set identifier of bulk
*              data
MQRMH_FORMAT   DS    CL8   Format name of bulk data
MQRMH_FLAGS    DS    F     Reference message flags
MQRMH_OBJECTTYPE DS    CL8  Object type
MQRMH_OBJECTINSTANCEID DS    XL24 Object instance identifier
MQRMH_SRCENVLENGTH DS    F   Length of source environment data
MQRMH_SRCENVOFFSET DS    F   Offset of source environment data
MQRMH_SRCNAMELENGTH DS    F   Length of source object name
MQRMH_SRCNAMEOFFSET DS    F   Offset of source object name
MQRMH_DESTENVLENGTH DS    F   Length of destination environment
*              data
MQRMH_DESTENVOFFSET DS    F   Offset of destination environment
*              data
MQRMH_DESTNAMELENGTH DS    F   Length of destination object name
MQRMH_DESTNAMEOFFSET DS    F   Offset of destination object name
MQRMH_DATALOGICALLLENGTH DS    F   Length of bulk data
MQRMH_DATALOGICALOFFSET DS    F   Low offset of bulk data
MQRMH_DATALOGICALOFFSET2 DS    F   High offset of bulk data
*
MQRMH_LENGTH   EQU    *-MQRMH
                ORG    MQRMH
MQRMH_AREA     DS    CL(MQRMH_LENGTH)

```

### Declaração do Visual Basic para MQRMH.

```

Type MQRMH
  StrucId      As String*4 'Structure identifier'
  Version      As Long     'Structure version number'
  StrucLength  As Long     'Total length of MQRMH, including'
                    'strings at end of fixed fields, but'
                    'not the bulk data'
  Encoding     As Long     'Numeric encoding of bulk data'
  CodedCharSetId As Long   'Character set identifier of bulk data'
  Format       As String*8 'Format name of bulk data'
  Flags       As Long     'Reference message flags'
  ObjectType   As String*8 'Object type'
  ObjectInstanceID As MQBYTE24 'Object instance identifier'
  SrcEnvLength As Long     'Length of source environment data'
  SrcEnvOffset As Long     'Offset of source environment data'
  SrcNameLength As Long    'Length of source object name'
  SrcNameOffset As Long    'Offset of source object name'
  DestEnvLength As Long    'Length of destination environment'
                    'data'
  DestEnvOffset As Long    'Offset of destination environment'
                    'data'
  DestNameLength As Long   'Length of destination object name'
  DestNameOffset As Long   'Offset of destination object name'
  DataLogicalLength As Long 'Length of bulk data'
  DataLogicalOffset As Long 'Low offset of bulk data'
  DataLogicalOffset2 As Long 'High offset of bulk data'
End Type

```

### **StrucId (MQCHAR4) para MQRMH.**

Este é o identificador da estrutura do cabeçalho da mensagem de referência. Ele é sempre um campo de entrada Seu valor é MQRMH\_STRUC\_ID.

O valor deve ser:.

### **MQRMH\_STRUC\_ID**

Identificador para a estrutura do cabeçalho da mensagem de referência.

Para a linguagem de programação C, a constante `MQRMH_STRUC_ID_ARRAY` também é definida. Ele tem o mesmo valor que `MQRMH_STRUC_ID`, mas é uma matriz de caracteres em vez de uma sequência.

### ***Versão (MQLONG) para MQRMH***

O número da versão da estrutura O valor deve ser:

#### **`MQRMH_VERSION_1`**

Version-1 estrutura do cabeçalho da mensagem de referência.

A constante a seguir especifica o número da versão atual:

#### **`MQRMH_CURRENT_VERSION`**

A versão atual da estrutura do cabeçalho da mensagem de referência

O valor inicial desse campo é `MQRMH_VERSION_1`.

### ***StrucLength (MQLONG) para MQRMH***

O comprimento total de `MQRMH`, incluindo sequências no término de campos fixos, mas não os dados em massa

O valor inicial deste campo é zero.

### ***Codificação (MQLONG) para MQRMH***

Isso especifica a codificação numérica dos dados em massa; ela não se aplica a dados numéricos na própria estrutura `MQRMH`

Na chamada `MQPUT` ou `MQPUT1`, o aplicativo deve configurar esse campo para o valor apropriado aos dados.

O valor inicial desse campo é `MQENC_NATIVE`.

### ***CodedCharSetId (MQLONG) para MQRMH***

Isso especifica o identificador do conjunto de caracteres dos dados em massa; ele não se aplica aos dados de caracteres na própria estrutura `MQRMH`

Na chamada `MQPUT` ou `MQPUT1`, o aplicativo deve configurar esse campo para o valor apropriado aos dados. O seguinte valor especial pode ser usado:

#### **`MQCCSI_INHERIT`**

Os dados de caracteres nos dados que seguem esta estrutura estão no mesmo conjunto de caracteres que esta estrutura

O gerenciador de filas altera esse valor na estrutura enviada na mensagem para o identificador do conjunto de caractere real da estrutura.. Desde que não ocorra nenhum erro, o valor `MQCCSI_INHERIT` não será retornado pela chamada `MQGET`

Não use `MQCCSI_INHERIT` se o valor do campo `PutApp1Type` em `MQMD` for `MQAT_BROKER`

Esse valor é suportado nos seguintes ambientes:

-  AIX
-  IBM i
-  Linux
-  Windows

e para clientes IBM MQ conectados a esses sistemas.

O valor inicial desse campo é `MQCCSI_UNDEFINED`.

### **Formato (MQCHAR8) para MQRMH**

Isso especifica o nome do formato dos dados em massa

Na chamada MQPUT ou MQPUT1 , o aplicativo deve configurar esse campo para o valor apropriado aos dados. As regras para codificação desse campo são as mesmas que aquelas para o campo *Format* no MQMD.

O valor inicial desse campo é MQFMT\_NONE.

### **Sinalizadores (MQLONG) para MQRMH**

Estes são sinalizadores de mensagem de referência Os sinalizadores a seguir são definidos:

#### **MQRMHF\_LAST**

Esse sinalizador indica que a mensagem de referência representa ou contém a última parte do objeto referido.

#### **MQRMHF\_NOT\_LAST**

A mensagem de referência não contém ou representa última parte do objeto. MQRMHF\_NOT\_LAST ajuda a documentação do programa. Não se pretende que essa opção seja usada com qualquer outra, mas como seu valor é zero, tal uso não pode ser detectado.

O valor inicial desse campo é MQRMHF\_NOT\_LAST.

### **ObjectType (MQCHAR8) para MQRMH**

Esse é um nome que a saída de mensagem pode usar para reconhecer os tipos de mensagens de referência que ela suporta O nome deve estar em conformidade com as mesmas regras que o campo *Format* , consulte [“Formato \(MQCHAR8\) para MQRMH”](#) na página 575

O valor inicial deste campo é 8 espaços em branco.

### **ObjectInstance(MQBYTE24) para MQRMH.**

Utilize este campo para identificar uma instância específica de um objeto Se não for necessário, configure-o para o valor a seguir:

#### **MQOII\_NONE**

Nenhum identificador de instância do objeto especificado. O valor é binário zero para o comprimento do campo.

Para a linguagem de programação C, a constante MQOII\_NONE\_ARRAY também é definida; isso tem o mesmo valor que MQOII\_NONE, mas é uma matriz de caracteres em vez de uma cadeia.

O comprimento desse campo é fornecido por MQ\_OBJECT\_INSTANCE\_ID\_LENGTH O valor inicial desse campo é MQOII\_NONE.

### **SrcEnvComprimento (MQLONG) para MQRMH**

O comprimento dos dados do ambiente de origem Se esse campo for zero, não haverá dados do ambiente de origem e *SrcEnvOffset* será ignorado.

O valor inicial desse campo é 0.

### **SrcEnvOffset (MQLONG)**

Esse campo especifica o deslocamento dos dados do ambiente de origem do início da estrutura MQRMH. Os dados do ambiente de origem podem ser especificados pelo criador da mensagem de referência, se esses dados forem conhecidos pelo criador.. Por exemplo, no Windows os dados do ambiente de origem podem ser o caminho do diretório do objeto contendo os dados em massa. No entanto, se o criador não souber os dados do ambiente de origem, a saída de mensagem fornecida pelo usuário deverá determinar quaisquer informações de ambiente necessárias..

O comprimento dos dados do ambiente de origem é fornecido por *SrcEnvLength* ; se esse comprimento for zero, não haverá dados do ambiente de origem e *SrcEnvOffset* será ignorado. Se presentes, os

dados do ambiente de origem devem residir completamente dentro de *StrucLength* bytes do início da estrutura.

Os aplicativos não devem assumir que os dados do ambiente sejam iniciados imediatamente após o último campo fixo da estrutura, ou que sejam contíguos com qualquer um dos dados endereçados pelos campos *SrcNameOffset*, *DestEnvOffset* e *DestNameOffset*

O valor inicial desse campo é 0.

### **Comprimento do SrcName(MQLONG) para MQRMH**

O comprimento do nome do objeto de origem. Se esse campo for zero, não haverá nome do objeto de origem e *SrcNameOffset* será ignorado.

O valor inicial desse campo é 0.

### **SrcNameOffset (MQLONG) para MQRMH**

Este campo especifica o deslocamento do nome do objeto de origem do início da estrutura MQRMH. O nome do objeto de origem pode ser especificado pelo criador da mensagem de referência, se esses dados forem conhecidos pelo criador.. No entanto, se o criador não souber o nome do objeto de origem, a saída de mensagem fornecida pelo usuário deverá identificar o objeto a ser acessado

O comprimento do nome do objeto de origem é fornecido por *SrcNameLength* ; se esse comprimento for zero, não haverá nome do objeto de origem e *SrcNameOffset* será ignorado. Se presente, o nome do objeto de origem deve residir completamente dentro de *StrucLength* bytes do início da estrutura.

Os aplicativos não devem assumir que o nome do objeto de origem seja contíguo com qualquer um desses dados endereçados pelos campos *SrcEnvOffset*, *DestEnvOffset* e *DestNameOffset*

O valor inicial desse campo é 0.

### **DestEnvComprimento (MQLONG) para MQRMH**

Este é o comprimento dos dados do ambiente de destino. Se esse campo for zero, não haverá dados do ambiente de destino e *DestEnvOffset* será ignorado.

### **DestEnvOffset (MQLONG) para MQRMH**

Este campo especifica o deslocamento dos dados do ambiente de destino do início da estrutura MQRMH. Os dados do ambiente de destino podem ser especificados pelo criador da mensagem de referência, se esses dados forem conhecidos pelo criador. Por exemplo, no Windows , os dados do ambiente de destino podem ser o caminho do diretório do objeto no qual os dados em massa devem ser armazenados. No entanto, se o criador não souber os dados do ambiente de destino, será de responsabilidade da saída de mensagem fornecida pelo usuário determinar quaisquer informações de ambiente necessárias

O comprimento dos dados do ambiente de destino é fornecido pelo *DestEnvLength* ; se esse comprimento for zero, não haverá dados do ambiente de destino e *DestEnvOffset* será ignorado. Se presentes, os dados do ambiente de destino devem residir completamente dentro de *StrucLength* bytes do início da estrutura.

Os aplicativos não devem assumir que os dados do ambiente de destino sejam contíguos com qualquer um dos dados endereçados pelos campos *SrcEnvOffset*, *SrcNameOffset* e *DestNameOffset*

O valor inicial desse campo é 0.

### **Comprimento do DestName(MQLONG) para MQRMH**

O comprimento do nome do objeto de destino.. Se esse campo for zero, não haverá nome do objeto de destino e *DestNameOffset* será ignorado.

### **DestNameOffset (MQLONG) para MQRMH**

Este campo especifica o deslocamento do nome do objeto de destino do início da estrutura MQRMH. O nome do objeto de destino pode ser especificado pelo criador da mensagem de referência, se esses

dados forem conhecidos pelo criador. No entanto, se o criador não souber o nome do objeto de destino, será responsabilidade da saída de mensagem fornecida pelo usuário identificar o objeto a ser criado ou modificado..

O comprimento do nome do objeto de destino é fornecido por *DestNameLength* ; se esse comprimento for zero, não haverá nome de objeto de destino e *DestNameOffset* será ignorado. Se presente, o nome do objeto de destino deve residir completamente dentro de *StrucLength* bytes do início da estrutura.

Os aplicativos não devem assumir que o nome do objeto de destino seja contíguo a qualquer um dos dados endereçados pelos campos *SrcEnvOffset*, *SrcNameOffset* e *DestEnvOffset*

O valor inicial desse campo é 0.

### **Comprimento do DataLogical(MQLONG) para MQRMH**

O campo *DataLogicalLength* especifica o comprimento dos dados em massa referenciados pela estrutura MQRMH.

Se os dados em massa estiverem realmente presentes na mensagem, os dados começarão em um deslocamento de *StrucLength* bytes do início da estrutura MQRMH. O comprimento da mensagem inteira menos *StrucLength* fornece o comprimento dos dados em massa presentes.

Se os dados estiverem presentes na mensagem, *DataLogicalLength* especifica a quantidade desses dados que é relevante. O caso normal é que *DataLogicalLength* tenha o mesmo valor que o comprimento dos dados presentes na mensagem.

Se a estrutura MQRMH representar os dados restantes no objeto (a partir do deslocamento lógico especificado), será possível usar o valor zero para *DataLogicalLength*, desde que os dados em massa não estejam realmente presentes na mensagem.

Se nenhum dado estiver presente, o final de MQRMH coincide com o final da mensagem.

O valor inicial desse campo é 0.

### **DataLogicalDeslocamento (MQLONG) para MQRMH**

Este campo especifica o deslocamento baixo dos dados em massa do início do objeto do qual os dados em massa fazem parte. O deslocamento dos dados em massa do início do objeto é chamado de *deslocamento lógico*. Esse não é o deslocamento físico dos dados em massa do início da estrutura MQRMH; esse deslocamento é fornecido pelo *StrucLength*.

Para permitir que objetos grandes sejam enviados usando mensagens de referência, o deslocamento lógico é dividido em dois campos e o deslocamento lógico real é fornecido pela soma desses dois campos:

- *DataLogicalOffset* representa o restante obtido quando o deslocamento lógico é dividido por 1 000 000 000. É, portanto, um valor no intervalo de 0 a 999 999 999.
- *DataLogicalOffset2* representa o resultado obtido quando o deslocamento lógico é dividido por 1 000 000 000. É, portanto, o número de múltiplos completos de 1 000 000 000 que existem no deslocamento lógico. O número de múltiplos está no intervalo de 0 a 999 999 999.

O valor inicial desse campo é 0.

### **DataLogicalOffset2 (MQLONG) para MQRMH**

Este campo especifica o alto deslocamento dos dados em massa do início do objeto do qual os dados em massa fazem parte. É um valor no intervalo de 0 a 999 999 999. Consulte *DataLogicalOffset* para obter detalhes.

O valor inicial desse campo é 0.

## MQRR-Registro de Resposta.

Use a estrutura MQRR para receber o código de conclusão e o código de razão resultantes da operação open ou put para uma única fila de destino, quando o destino for uma lista de distribuição. MQRR é uma estrutura de saída para as chamadas MQOPEN, MQPUT e MQPUT1 .

### Disponibilidade

A estrutura MQRR está disponível nas plataformas a seguir:

-  AIX
-  IBM i
-  Linux
-  Windows

e para clientes IBM MQ conectados a esses sistemas.

### Conjunto de caracteres e codificação

Os dados em MQRR devem estar no conjunto de caracteres fornecido pelo atributo do gerenciador de fila do **CodedCharSetId** e pela codificação do gerenciador de fila local fornecido por MQENC\_NATIVE. No entanto, se o aplicativo estiver em execução como um cliente MQI MQ , a estrutura deverá estar no conjunto de caracteres e na codificação do cliente.

### Uso

Fornecendo uma matriz dessas estruturas nas chamadas MQOPEN e MQPUT ou na chamada MQPUT1 , é possível determinar os códigos de conclusão e os códigos de razão para todas as filas em uma lista de distribuição quando o resultado da chamada é combinado, ou seja, quando a chamada é bem-sucedida para algumas filas na lista, mas falha para outras. O código de razão MQRC\_MULTIPLE\_REASON da chamada indica que os registros de resposta (se fornecidos pelo aplicativo) foram configurados pelo gerenciador de filas..

### Campos

**Nota:** Na tabela a seguir, os campos são agrupados por uso e não alfabeticamente. Os tópicos filhos seguem a mesma sequência.

<i>Tabela 524. Campos em MQRR</i>		
Nome e descrição do campo	Nome da constante	Valor inicial (se houver) da constante
CompCode (código de conclusão para a fila)	MQCC_OK	0
Razão (código de razão para fila)	MQRC_NONE	0

**Notas:**

1. Na linguagem de programação C, a variável macroMQRR\_DEFAULT contém os valores listados na tabela. Use-o da seguinte maneira para fornecer valores iniciais para os campos na estrutura:

```
MQRR MyRR = {MQRR_DEFAULT};
```

## Declarações de idiomas

### Declaração C para MQRR

```
typedef struct tagMQRR MQRR;
struct tagMQRR {
    MQLONG  CompCode; /* Completion code for queue */
    MQLONG  Reason; /* Reason code for queue */
};
```

### Declaração COBOL para MQRR

```
** MQRR structure
10 MQRR.
** Completion code for queue
15 MQRR-COMPCODE PIC S9(9) BINARY.
** Reason code for queue
15 MQRR-REASON PIC S9(9) BINARY.
```

### Declaração PL/I para MQRR

```
dcl
  1 MQRR based,
  3 CompCode fixed bin(31), /* Completion code for queue */
  3 Reason fixed bin(31); /* Reason code for queue */
```

### Declaração Visual Basic para MQRR

```
Type MQRR
  CompCode As Long 'Completion code for queue'
  Reason As Long 'Reason code for queue'
End Type
```

## **CompCode (MQLONG) para MQRR**

Este é o código de conclusão resultante da operação open ou put para a fila com o nome que foi especificado pelo elemento correspondente na matriz de estruturas MQOR fornecidas na chamada MQOPEN ou MQPUT1 .

Este é sempre um campo de saída O valor inicial desse campo é MQCC\_OK..

## **Razão (MQLONG) para MQRR**

Este é o código de razão resultante da operação open ou put para a fila com o nome que foi especificado pelo elemento correspondente na matriz de estruturas MQOR fornecidas na chamada MQOPEN ou MQPUT1 .


Este é sempre um campo de saída O valor inicial desse campo é MQRC\_NONE.

## **Opções de configuração de MQSCO-SSL/TLS**

A estrutura MQSCO, em conjunto com os campos TLS na estrutura MQCD, permite que um aplicativo em execução como um IBM MQ MQI client para especificar opções de configuração que controlam o uso de TLS para a conexão do cliente quando o protocolo de canal é TCP/IP. A estrutura é um parâmetro de saída na chamada MQCONN.

## **Disponibilidade**

A estrutura MQSCO está disponível nos clientes a seguir:

-  AIX
-  IBM i

-  Linux
-  Windows

Se o protocolo do canal para o canal cliente não for TCP/IP, a estrutura MQSCO será ignorada.

## Conjunto de caracteres e codificação

Os dados no MQSCO devem estar no conjunto de caracteres fornecido pelo atributo do gerenciador de fila do **CodedCharSetId** e na codificação do gerenciador de filas locais fornecido pelo MQENC\_NATIVE







## Campos

**Nota:** Na tabela a seguir, os campos são agrupados por uso e não alfabeticamente. Os tópicos filhos seguem a mesma sequência.

<i>Tabela 525. Campos em MQSCO</i>		
Nome e descrição do campo	Nome da constante	Valor inicial (se houver) da constante
<u>StrucId</u> (identificador de estrutura)	MQSCO_STRUC_ID	'SCO→'
<u>Versão</u> (número da versão da estrutura).	MQSCO_CURRENT_VERSION	1
<u>KeyRepository</u> (local do repositório de chaves)	Nenhum	Cadeia nula ou espaços em branco
<u>CryptoHardware</u> (detalhes do hardware de criptografia)	Nenhum	Cadeia nula ou espaços em branco
<u>AuthInfoRecCount</u> (número de registros MQAIR presentes)	Nenhum	0
<u>AuthInfoRecOffset</u> (deslocamento do primeiro registro MQAIR do início de MQSCO)	Nenhum	0
<u>AuthInfoRecPtr</u> (endereço do primeiro registro MQAIR)	Nenhum	Ponteiro nulo ou bytes nulos
<b>Nota:</b> Os dois campos a seguir serão ignorados se <i>Version</i> for menor que MQSCO_VERSION_2..		
<u>KeyResetCount</u> (contagem de reconfiguração de chave secreta TLS)	MQSCO_RESET_COUNT_DEFAULT	0
<u>“FipsRequired (MQLONG) para MQSCO” na página 586</u> (use algoritmos criptográficos certificados pelo FIPS no IBM MQ).	MQSSL_FIPS_NO	0
<b>Nota:</b> Os dois campos a seguir serão ignorados se <i>Version</i> for menor que MQSCO_VERSION_3..		
<u>EncryptionPolicySuiteB</u> (use apenas algoritmos criptográficos Suite B)	MQ_SUITE_B_NONE, MQ_SUITE_B_NOT_AVAILABLE, MQ_SUITE_B_NOT_AVAILABLE, MQ_SUITE_B_NOT_AVAILABLE	1, 0, 0, 0
<b>Nota:</b> Os dois campos a seguir serão ignorados se <i>Version</i> for menor que MQSCO_VERSION_4..		



Tabela 525. Campos em MQSCO (continuação)

Nome e descrição do campo	Nome da constante	Valor inicial (se houver) da constante
PolíticaCertificateVal (política de validação de certificado)	MQ_CERT_VAL_POLICY_DEFAULT	0
<b>Nota:</b> Os dois campos a seguir serão ignorados se <i>Version</i> for menor que MQSCO_VERSION_5..		
CertificateLabel (detalha o rótulo certificado que está sendo usado)	Nenhum	Cadeia nula ou espaços em branco
<b>Nota:</b> Os campos restantes serão ignorados se <i>Version</i> for menor que MQSCO_VERSION_6..		
  KeyRepoPasswordPtr (endereço da senha do repositório de chaves TLS)	Nenhum	Ponteiro nulo ou bytes nulos
  KeyRepoPasswordOffset (compensação de senha do repositório de chaves TLS)	Nenhum	0
  KeyRepoPasswordLength (comprimento da senha do repositório de chaves TLS)	Nenhum	0

**Notas:**

1. O símbolo – representa um único caractere em branco.
2. Na linguagem de programação C, a variável macroMQSCO\_DEFAULT contém os valores listados na tabela.. Use-o da seguinte maneira para fornecer valores iniciais para os campos na estrutura:

```
MQSCO MySCO = {MQSCO_DEFAULT};
```

**Declarações de idiomas**

Declaração C para MQSCO

```
typedef struct tagMQSCO MQSCO;
struct tagMQSCO {
    MCHAR4      StrucId;                /* Structure identifier */
    MQLONG      Version;                /* Structure version number */
    MCHAR256    KeyRepository;          /* Location of TLS key */
                                          /* repository */
    MCHAR256    CryptoHardware;         /* Cryptographic hardware */
                                          /* configuration string */
    MQLONG      AuthInfoRecCount;       /* Number of MQAIR records */
                                          /* present */
    MQLONG      AuthInfoRecOffset;      /* Offset of first MQAIR */
                                          /* record from start of */
                                          /* MQSCO structure */
    PMQAIR      AuthInfoRecPtr;         /* Address of first MQAIR */
                                          /* record */
    /* Ver:1 */
    MQLONG      KeyResetCount;          /* Number of unencrypted */
                                          /* bytes sent/received */
                                          /* before secret key is */
                                          /* reset */
    MQLONG      FipsRequired;           /* Using FIPS-certified */
    /* Ver:2 */
                                          /* algorithms */
    MQLONG      EncryptionPolicySuiteB[4]; /* Use only Suite B */
    /* Ver:3 */
    MQLONG      CertificateValPolicy;   /* cryptographic algorithms */
                                          /* Certificate validation */
}
```

```

/* Ver:4 */
MQCHAR64 CertificateLabel; /* Certificate label */
/* Ver:5 */
MQPTR KeyRepoPasswordPtr; /* Address of key */
/* repository password */
MQLONG KeyRepoPasswordOffset; /* Offset of key repository */
/* password */
MQLONG KeyRepoPasswordLength; /* Length of key repository */
/* password */
/* Ver:6 */
};

```

## Declaração COBOL para MQSCO

```

** MQSCO structure
10 MQSCO.
** Structure identifier
15 MQSCO-STRUCID PIC X(4).
** Structure version number
15 MQSCO-VERSION PIC S9(9) BINARY.
** Location of TLS key repository
15 MQSCO-KEYREPOSITORY PIC X(256).
** Cryptographic hardware configuration string
15 MQSCO-CRYPTOHARDWARE PIC X(256).
** Number of MQAIR records present
15 MQSCO-AUTHINFORECCOUNT PIC S9(9) BINARY.
** Offset of first MQAIR record from start of MQSCO structure
15 MQSCO-AUTHINFORECOFFSET PIC S9(9) BINARY.
** Address of first MQAIR record
15 MQSCO-AUTHINFORECPTR POINTER.
** Version 1 **
** Number of unencrypted bytes sent/received before secret key is
** reset
15 MQSCO-KEYRESETCOUNT PIC S9(9) BINARY.
** Using FIPS-certified algorithms
15 MQSCO-FIPSREQUIRED PIC S9(9) BINARY.
** Version 2 **
** Use only Suite B cryptographic algorithms
15 MQSCO-ENCRYPTIONPOLICYSUITEB PIC S9(9) BINARY OCCURS 4.
** Version 3 **
** Certificate validation policy setting
15 MQSCO-CERTIFICATEVALPOLICY PIC S9(9) BINARY.
** Version 4 **
** SSL/TLS certificate label
15 MQSCO-CERTIFICATELABEL PIC X(64).
** Version 5 **
** Add padding to ensure that pointers start on correct
** boundaries
15 FILLER PIC S9(9) BINARY VALUE 0.
** Address of key repository password
15 MQSCO-KEYREPOPASSWORDPTR POINTER.
** Offset of key repository password
15 MQSCO-KEYREPOPASSWORDOFFSET PIC S9(9) BINARY.
** Length of key repository password
15 MQSCO-KEYREPOPASSWORDLENGTH PIC S9(9) BINARY.
** Version 6 **

```

## Declaração PL/I para MQSCO

```

dcl
1 MQSCO based,
3 StrucId char(4), /* Structure identifier */
3 Version fixed bin(31), /* Structure version number */
3 KeyRepository char(256), /* Location of TLS key
repository */
3 CryptoHardware char(256), /* Cryptographic hardware
configuration string */
3 AuthInfoRecCount fixed bin(31), /* Number of MQAIR records
present */
3 AuthInfoRecOffset fixed bin(31), /* Offset of first MQAIR record
from start of MQSCO structure */
3 AuthInfoRecPtr pointer, /* Address of first MQAIR record */
3 KeyResetCount fixed bin(31), /* Key reset count */
/* Version 1 */
3 FipsRequired fixed bin(31), /* FIPS required */
/* Version 2 */

```

```

3 EncryptionPolicySuiteB (4) fixed bin(31), /* Suite B encryption policy */
/* Version 3 */
3 CertificateValPolicy      fixed bin(31), /* Certificate validation policy */
/* Version 4 */
3 CertificateLabel          char(64),      /* SSL/TLS certificate label */
/* Version 5 */
3 KeyRepoPasswordPtr        pointer,      /* Address of key repository
password */
3 KeyRepoPasswordOffset     fixed bin(31), /* Offset of key repository
password */
3 KeyRepoPasswordLength     fixed bin(31); /* Length of key repository
password */
/* Version 6 */

```

## Declaração do Visual Basic para MQSCO

```

Type MQSCO
  StrucId      As String*4      'Structure identifier'
  Version      As Long          'Structure version number'
  KeyRepository As String*256   'Location of TLS key repository'
  CryptoHardware As String*256 'Cryptographic hardware configuration'
  AuthInfoRecCount As Long      'Number of MQAIR records present'
  AuthInfoRecOffset As Long     'Offset of first MQAIR record from'
  AuthInfoRecPtr As MQPTR      'Address of first MQAIR record'
  KeyResetCount As Long        'Number of unencrypted bytes sent/received before secret key
is reset'
  'Version 1'
  FipsRequired As Long         'Mandatory FIPS CipherSpecs?'
  'Version 2'
End Type

```

### Referências relacionadas

“MQCNO-Opções de conexão” na página 324

A estrutura MQCNO permite que o aplicativo especifique opções relacionadas à conexão com o gerenciador de filas.. A estrutura é um parâmetro de entrada / saída na chamada MQCONN.

### **StrucId (MQCHAR4) para MQSCO**

Esse é o identificador de estrutura da estrutura de opções de configuração SSL/TLS. Ele é sempre um campo de entrada Seu valor é MQSCO\_STRUC\_ID.

O valor deve ser:.

#### **MQSCO\_STRUC\_ID**

Identificador para a estrutura de opções de configuração SSL/TLS.

Para a linguagem de programação C, a constante MQSCO\_STRUC\_ID\_ARRAY também é definida. Isso tem o mesmo valor que MQSCO\_STRUC\_ID, mas é uma matriz de caracteres em vez de uma sequência.

### **Versão (MQLONG) para MQSCO**

Este é o número da versão da estrutura; o valor deve ser:

#### **MQSCO\_VERSION\_1**

Version-1 Estrutura de opções de configuração TLS.

#### **MQSCO\_VERSION\_2**

Estrutura de opções de configuração do TLS Version-2

#### **MQSCO\_VERSION\_3**

Version-3 Estrutura de opções de configuração TLS.

#### **MQSCO\_VERSION\_4**

Version-4 Estrutura de opções de configuração TLS.

#### **MQSCO\_VERSION\_5**

Estrutura de opções de configuração do TLS Version-5

V 9.3.0

V 9.3.0

## **MQSCO\_VERSION\_6**

Estrutura de opções de configuração do TLS Version-6

A constante a seguir especifica o número da versão atual:

### **MQSCO\_CURRENT\_VERSION**

Versão atual da estrutura de opções de configuração do TLS

Isto é sempre um campo de entrada. O valor inicial desse campo é MQSCO\_VERSION\_1.

Multi

### **KeyRepository (MQCHAR256) para MQSCO**

Esse campo é relevante apenas para IBM MQ MQI clients em execução em IBM i, AIX, Linux, and Windows sistemas.... Ele especifica o local do arquivo de banco de dados de chave no qual as chaves e os certificados são armazenados. Se o sufixo do arquivo não for especificado, um sufixo .kdb será incluído automaticamente.

Cada arquivo do banco de dados de chave pode ter um *arquivo stash de senha* associado. O arquivo stash contém senhas codificadas que são usadas para permitir acesso programático ao banco de dados de chaves. O arquivo stash de senha deve residir no mesmo diretório e ter a mesma raiz de arquivo que o banco de dados de chave e deve terminar com o sufixo .sth

Por exemplo, se o arquivo do banco de dados de chave for /xxx/yyy/key.kdb, o arquivo stash de senha deverá ser /xxx/yyy/key.sth, em que xxx e yyy representarão nomes de diretórios

V 9.3.0

V 9.3.0

A senha do banco de dados chave também pode ser especificada usando o campo *KeyRepoPasswordPtr* ou *KeyRepoPasswordOffset* da estrutura MQSCO.

Se o valor for menor que o comprimento do campo, termine o valor com um caractere nulo ou preencha-o com espaços em branco até o comprimento do campo. O valor não é verificado; se houver um erro ao acessar o repositório de chaves, a chamada falhará com o código de razão MQRC\_KEY\_REPOSITORY\_ERROR...

Para executar uma conexão TLS a partir de um IBM MQ MQI client, configure *KeyRepository* para um nome de arquivo de banco de dados de chaves válido

Este é um campo de entrada. O comprimento desse campo é fornecido por MQ\_SSL\_KEY\_REPOSITORY\_LENGTH. O valor inicial desse campo é a cadeia nula em C e caracteres em branco em outras linguagens de programação.

### **CryptoHardware (MQCHAR256) para MQSCO**

Este campo fornece detalhes de configuração para o hardware criptográfico conectado ao sistema do cliente

Configure o campo para uma sequência do seguinte formato ou deixe-o em branco ou nulo:

```
GSK_PKCS11=the PKCS #11 driver path and file name;the PKCS #11 token label;the PKCS #11 token password;symmetric cipher setting;
```

Para usar o hardware de criptografia que está em conformidade com a interface PKCS #11, por exemplo, IBM 4960 ou IBM 4764, o caminho do driver PKCS #11, o rótulo do token PKCS #11 e as sequências de senha do token PKCS #11 devem ser especificados, cada um finalizado por ponto e vírgula.

O caminho do driver PKCS #11 é um caminho absoluto para a biblioteca compartilhada que fornece suporte para a placa PKCS #11. O nome do arquivo do driver PKCS #11 é o nome da Biblioteca Compartilhada. Um exemplo do valor necessário para o nome do caminho e do arquivo PKCS #11 é:

```
/usr/lib/pkcs11/PKCS11_API.so
```

O rótulo do token PKCS #11 deve corresponder ao rótulo com o qual seu hardware foi configurado.

Se nenhuma configuração de hardware de criptografia for necessária, configure o campo como em branco ou nulo.

Se o valor for menor que o comprimento do campo, termine o valor com um caractere nulo ou preencha-o com espaços em branco até o comprimento do campo. Se o valor não for válido ou levar a uma falha quando usado para configurar o hardware criptográfico, a chamada falhará com o código de razão MQRD\_CRYPTD\_HARDWARE\_ERROR.

Este é um campo de entrada. O comprimento desse campo é fornecido por MQ\_SSL\_CRYPTD\_HARDWARE\_LENGTH O valor inicial desse campo é a cadeia nula em C e caracteres em branco em outras linguagens de programação.

### ***AuthInfoRecCount (MQLONG) para MQSCO***

Este é o número de registros de informações sobre autenticação (MQAIR) endereçados pelos campos *AuthInfoRecPtr* ou *AuthInfoRecOffset* Para obter mais informações, consulte “MQAIR-Registro de informações sobre autenticação.” na página 276.. O valor deverá ser zero ou maior. Se o valor não for válido, a chamada falha com o código de razão MQRD\_AUTH\_INFO\_REC\_COUNT\_ERROR.

Este é um campo de entrada. O valor inicial desse campo é 0.

### ***AuthInfoRecOffset (MQLONG) para MQSCO***

Este é o deslocamento em bytes do primeiro registro de informações sobre autenticação a partir do início da estrutura MQSCO. O deslocamento pode ser positivo ou negativo O campo será ignorado se *AuthInfoRecCount* for zero.

É possível usar *AuthInfoRecOffset* ou *AuthInfoRecPtr* para especificar os registros MQAIR, mas não ambos; consulte a descrição do campo *AuthInfoRecPtr* para obter detalhes..

Este é um campo de entrada. O valor inicial desse campo é 0.

### ***AuthInfoRecPtr (PMQAIR) para MQSCO***

Este é o endereço do primeiro registro de informações sobre autenticação O campo será ignorado se *AuthInfoRecCount* for zero.

É possível fornecer a matriz de registros MQAIR de uma de duas maneiras:

- Usando o campo de ponteiro *AuthInfoRecPtr* .

Nesse caso, o aplicativo pode declarar uma matriz de registros MQAIR que é separada da estrutura MQSCO e configurar *AuthInfoRecPtr* para o endereço da matriz

Considere o uso do *AuthInfoRecPtr* para linguagens de programação que suportam o tipo de dados do ponteiro de uma maneira que seja móvel para ambientes diferentes (por exemplo, a linguagem de programação C)

- Usando o campo de deslocamento *AuthInfoRecOffset*

Nesse caso, o aplicativo deve declarar uma estrutura composta contendo um MQSCO seguido pela matriz de registros MQAIR e configurar *AuthInfoRecOffset* para o deslocamento do primeiro registro na matriz do início da estrutura MQSCO. Assegure-se de que esse valor esteja correto e tenha um valor que possa ser acomodado em um MQLONG (a linguagem de programação mais restritiva é COBOL, para o qual o intervalo válido é -999 999 999 a +999 999 999).

Considere o uso do *AuthInfoRecOffset* para linguagens de programação que não suportam o tipo de dados do ponteiro ou que implementam o tipo de dados do ponteiro de forma que não seja móvel para ambientes diferentes (por exemplo, a linguagem de programação COBOL).

Independentemente da técnica escolhida, apenas um de *AuthInfoRecPtr* e *AuthInfoRecOffset* pode ser usado; a chamada falhará com o código de razão MQRD\_AUTH\_INFO\_REC\_ERROR se ambos forem diferentes de zero.

Este é um campo de entrada. O valor inicial deste campo é o ponteiro nulo nas linguagens de programação que suportam ponteiros e, caso contrário, uma cadeia de bytes all-null.

**Nota:** Nas plataformas em que a linguagem de programação não suporta o tipo de dados do ponteiro, este campo é declarado como uma cadeia de bytes do comprimento apropriado.

### ***KeyResetContagem (MQLONG) para MQSCO***

Isso representa o número total de bytes não criptografados enviados e recebidos em uma conversa TLS antes da renegociação da chave secreta.

O número de bytes inclui informações de controle enviadas pelo MCA.

Se você especificar uma contagem de reconfiguração de chave secreta TLS no intervalo de 1 byte a 32 KB, os canais TLS usarão uma contagem de reconfiguração de chave secreta de 32 KB. Isso é para evitar o custo de processamento de reconfigurações de chave excessivas que ocorreriam para pequenos valores de reconfiguração de chave secreta TLS..

Este é um campo de entrada. O valor é um número no intervalo de 0 a 999 999 999, com um valor padrão de 0. Use um valor 0 para indicar que as chaves secretas nunca são renegociadas.

### ***FipsRequired (MQLONG) para MQSCO***

O IBM MQ pode ser configurado com o hardware de criptografia para que os módulos de criptografia usados sejam aqueles fornecidos pelo produto de hardware; eles podem ser certificados pelo FIPS para um nível específico, dependendo do produto de hardware de criptografia em uso. Use esse campo para especificar que apenas algoritmos-certificados FIPS serão usados se a criptografia for fornecida no software fornecido pelo IBM MQ

**Nota:** No AIX, Linux, and Windows, IBM MQ fornece conformidade FIPS 140-2 por meio do módulo criptográfico IBM Crypto for C (ICC) . O certificado deste módulo foi movido para o status Histórico. Os clientes devem visualizar o [IBM Crypto for C \(ICC\) certificado](#) e estar ciente de qualquer aviso fornecido pelo NIST Um módulo FIPS 140-3 de substituição está atualmente em andamento e seu status pode ser visualizado procurando por ele na [NIST CMVP modules in process list](#).

Quando o IBM MQ é instalado, uma implementação de criptografia TLS também é instalada, o que fornece alguns módulos certificados pelo FIPS

Os valores podem ser:

#### **MQSSL\_FIPS\_NO**

Esse é o valor-padrão. Quando configurado para este valor:

- Qualquer CipherSpec suportado em uma plataforma específica pode ser usado
- Se executado sem o uso de hardware de criptografia, o CipherSpecs será executado usando a criptografia certificada FIPS 140-2 nas plataformas IBM MQ .

Para obter uma lista de CipherSpecscertificados FIPS, consulte a tabela descrita em [Ativando CipherSpecs](#).

#### **MQSSL\_FIPS\_YES**

Quando configurado para esse valor, a menos que você esteja usando o hardware de criptografia para executar a criptografia, é possível ter certeza de que

- Somente algoritmos criptográficos certificados pelo FIPS podem ser usados no CipherSpec que se aplica a essa conexão do cliente
- As conexões do canal TLS de entrada e de saída serão bem-sucedidas apenas se determinadas Especificações de Cifra forem usadas.

Consulte [Ativando o CipherSpecs](#) para obter mais informações

**Nota:** Onde possível, se CipherSpecs somente FIPS forem configurados então o cliente MQI rejeita conexões que especificam um CipherSpec não FIPS com MQRC\_SSL\_INITIALIZATION\_ERROR. O IBM MQ não garante rejeitar todas essas conexões e é sua responsabilidade determinar se sua configuração do IBM MQ está com o padrão FIPS.

### ***EncryptionPolicySuiteB (MQLONG) para MQSCO.***

Este campo Especifica se a criptografia compatível com o Conjunto B é usada e qual nível de intensidade é empregado O valor pode ser um ou mais de:

- MQ\_SUITE\_B\_NONE

A criptografia compatível com o Conjunto B não é usada

- MQ\_SUITE\_B\_128\_BIT

A segurança de força do conjunto B de 128 bits é usada

- MQ\_SUITE\_B\_192\_BIT

A segurança de força do conjunto B de 192 bits é usada

**Nota:** Usar MQ\_SUITE\_B\_NONE com qualquer outro valor nesse campo é inválido.

### **Política CertificateVal(MQLONG) para MQSCO**

Este campo especifica qual tipo de política de validação de certificado é usado

O campo pode ser configurado para um dos seguintes valores:

#### **MQ\_CERT\_VAL\_POLICY\_ANY**

Aplique cada uma das políticas de validação de certificado suportadas pela biblioteca de soquetes seguros.. Aceite a cadeia de certificados se qualquer uma das políticas considerar a cadeia de certificados válida

#### **MQ\_CERT\_VAL\_POLICY\_RFC5280**

Aplique apenas a política de validação de certificado compatível com o RFC5280 Esta configuração fornece validação mais estrita do que a configuração ANY, mas rejeita alguns certificados digitais mais antigos.

O valor inicial desse campo é MQ\_CERT\_VAL\_POLICY\_ANY

### **CertificateLabel (MQCHAR64) para MQSCO**

Este campo fornece detalhes do rótulo do certificado que está sendo usado

IBM MQ inicializa o valor padrão para o campo *CertificateLabel* como espaços em branco.

Isto é interpretado no tempo de execução como o valor padrão e é compatível com versões anteriores.

Por exemplo, especificar uma versão MQSCO menor que 5.0 ou usar o valor padrão de espaços em branco para o campo *CertificateLabel* , usa o valor padrão preexistente de `ibmwebsphereuser_id`.

### **KeyRepoPasswordPtr (MQPTR) para MQSCO**

Este é o endereço em bytes da passphrase do repositório de chaves TLS

Este é um campo de entrada. O valor inicial deste campo é o ponteiro nulo nas linguagens de programação que suportam ponteiros e, caso contrário, uma cadeia de bytes all-null. Esse campo será ignorado se *Version* for menor que MQSCO\_VERSION\_6.

Esse campo é relevante apenas para IBM MQ MQI clients em execução em IBM i, AIX, Linux, and Windows sistemas....

A passphrase do repositório de chave pode ser especificada como uma sequência de texto simples ou uma passphrase que foi criptografada usando o utilitário **runmqicred** .

Se você fornecer uma passphrase criptografada, especifique a chave inicial que foi usada para criptografar a passphrase na estrutura MQCSP fornecida pelo mesmo aplicativo cliente.

A passphrase do repositório de chave especificada usando esse campo substitui qualquer passphrase do repositório de chave especificada usando a variável de ambiente `MQKEYRPWD` ou a propriedade `SSLKeyRepositoryPassword` na sub-rotina SSL do arquivo de configuração do cliente.

É possível usar `KeyRepoPasswordOffset` ou `KeyRepoPasswordPtr` para especificar a passphrase do repositório de chave, mas não ambos..

## Tarefas relacionadas

Fornecendo uma chave inicial para um cliente MQI do IBM MQ no AIX, Linux e Windows

[Protegendo senhas nos arquivos de configuração do componente IBM MQ](#)

## Referências relacionadas

[runmqicred \(proteger senhas do cliente IBM MQ\)](#)

[“InitialKeyPtr \(MQPTR\) para MQCSP” na página 350](#)

O endereço para a chave inicial para o sistema de proteção de senha

### **KeyRepoPasswordOffset (MQLONG) para MQSCO**

Este é o deslocamento em bytes da passphrase do repositório de chaves TLS do início da estrutura MQSCO. O deslocamento pode ser positivo ou negativo

É possível usar *KeyRepoPasswordOffset* ou *KeyRepoPasswordPtr* para especificar a passphrase do repositório de chave, mas não ambos.. Para obter mais informações, consulte a descrição do campo *KeyRepoPasswordPtr* ..

Este é um campo de entrada. O valor inicial desse campo é 0. Esse campo será ignorado se *Version* for menor que MQSCO\_VERSION\_6.

### **KeyRepoPasswordLength (MQLONG) para MQSCO**

Este é o comprimento da passphrase do repositório de chaves TLS..

No IBM i, o comprimento máximo da passphrase do repositório de chave é 128 caracteres. Se a passphrase do repositório de chaves for maior do que o comprimento máximo permitido, a conexão falhará com MQRC\_KEY\_REPOSITORY\_ERROR


Este é um campo de entrada. O valor inicial desse campo é 0. Esse campo será ignorado se *Version* for menor que MQSCO\_VERSION\_6.

## MQSD - Descritor de Assinatura

A estrutura MQSD é usada para especificar detalhes sobre a assinatura sendo feita. A estrutura é um parâmetro de entrada / saída na chamada do MQSUB. Para obter mais informações, consulte [Notas de uso MQSUB](#).

## Disponibilidade

A estrutura MQSD está disponível nas plataformas a seguir:

-  AIX
-  IBM i
-  Linux
-  Windows
-  z/OS

e para o IBM MQ MQI clients conectado a esses sistemas

## Versão

A versão atual do MQSD é MQSD\_VERSION\_1..

## Conjunto de caracteres e codificação

Os dados no MQSD devem estar no conjunto de caracteres fornecido pelo atributo do gerenciador de filas do **CodedCharSetId** e pela codificação do gerenciador de fila local fornecido por MQENC\_NATIVE



No entanto, se o aplicativo estiver em execução como um cliente MQI MQ , a estrutura deverá estar no conjunto de caracteres e na codificação do cliente.

## Assinaturas gerenciadas

Se um aplicativo não tiver necessidade específica de usar uma fila específica como o destino para aquelas publicações que correspondem a sua assinatura, ele poderá usar o recurso de assinatura gerenciada. Se um aplicativo optar por usar uma assinatura gerenciada, o gerenciador de filas informará o assinante sobre o destino para o qual as mensagens publicadas são enviadas, fornecendo uma manipulação do objeto como uma saída da chamada MQSUB. Para obter mais informações, consulte [Hobj \(MQHOBJ\)- entrada / saída](#).

Quando a assinatura é removida, o gerenciador de filas também se compromete a limpar mensagens que não foram recuperadas do destino gerenciado, nas seguintes situações:

- Quando a assinatura é removida-pelo uso de MQCLOSE com MQCO\_REMOVE\_SUB-e o Hobj gerenciado é fechado..
- Por meio implícito quando a conexão é perdida para um aplicativo usando uma assinatura não durável (MQSO\_NON\_DURABLE)
- Por expiração quando uma assinatura é removida porque expirou e o Hobj gerenciado é fechado.

Deve-se usar assinaturas gerenciadas com assinaturas não duráveis para que essa limpeza possa ocorrer e para que as mensagens para assinaturas não duráveis fechadas não ocupem espaço em seu gerenciador de filas. Assinaturas duráveis também podem usar destinos gerenciados.

## Campos

**Nota:** Na tabela a seguir, os campos são agrupados por uso e não alfabeticamente. Os tópicos filhos seguem a mesma sequência.

Nome e descrição do campo	Nome da constante	Valor inicial (se houver) da constante
<u>StrucId</u> (identificador de estrutura)	MQSD_STRUC_ID	'SD→→'
<u>Versão</u> (número da versão da estrutura).	MQSD_VERSION_1	1
<u>Opções</u> (opções).	MQSO_NON_DURABLE	0
<u>ObjectName</u> (nome do objeto)	Nenhum	Cadeia nula ou espaços em branco
<u>AlternateUserId</u> (ID do usuário alternativo)	Nenhum	Cadeia nula ou espaços em branco
<u>AlternateSecurityId</u> (ID de segurança alternativo)	MQSID_NONE	Nulos
<u>SubExpiry</u> (expiração da assinatura)	MQEI_UNLIMITED	-1
<u>ObjectString</u> (sequência de objetos).	Nenhum	Nomes e valores conforme definido para MQCHARV
<u>SubName</u> (nome da assinatura)	Nenhum	Nomes e valores conforme definido para MQCHARV
<u>SubUser</u> (dados do usuário de assinatura)	Nenhum	Nomes e valores conforme definido para MQCHARV
<u>SubCorrelSubCorrel</u> (ID de correlação de assinatura)	MQCI_NONE	Nulos

Nome e descrição do campo	Nome da constante	Valor inicial (se houver) da constante
<u>PubPriority</u> (prioridade de publicação)	MQPRI_PRIORITY_AS_Q_DEF	-3
<u>PubAccounting</u> (token de contabilidade de publicação)	MQACT_NONE	Nulos
<u>PubAppIdentityData</u> (dados de identidade do aplicativo de publicação)..	Nenhum	Cadeia nula ou espaços em branco
<u>SelectionString</u> (sequência fornecendo critérios de seleção)	Nenhum	Nomes e valores conforme definido para MQCHARV
<u>SubLevel</u> (nível de assinatura)	Nenhum	1
<u>ResObjectSequência</u> (nome do objeto longo)	Nenhum	Nomes e valores conforme definido para MQCHARV
<b>Notas:</b>		
<ol style="list-style-type: none"> <li>1. O símbolo – representa um único caractere em branco.</li> <li>2. O valor Sequência nula ou espaços em branco indica a sequência nula em C e caracteres em branco em outras linguagens de programação.</li> <li>3. Na linguagem de programação C, a variável macroMQSD_DEFAULT contém os valores listados na tabela. Ele pode ser usado da seguinte maneira para fornecer valores iniciais para os campos na estrutura: <div style="background-color: #f0f0f0; padding: 5px; margin-top: 10px;"> <pre>MQSD MySD = {MQSD_DEFAULT};</pre> </div> </li> </ol>		

## Declarações de idiomas

### Declaração C para MQSD

```
typedef struct tagMQSD MQSD;
struct tagMQSD {
    MQCHAR4    StrucId;           /* Structure identifier */
    MQLONG    Version;          /* Structure version number */
    MQLONG    Options;          /* Options associated with subscribing */
    MQCHAR48   ObjectName;      /* Object name */
    MQCHAR12  AlternateUserId;  /* Alternate user identifier */
    MQBYTE40  AlternateSecurityId; /* Alternate security identifier */
    MQLONG    SubExpiry;        /* Expiry of Subscription */
    MQCHARV   ObjectString;     /* Object Long name */
    MQCHARV   SubName;          /* Subscription name */
    MQCHARV   SubUserData;      /* Subscription User data */
    MQBYTE24  SubCorrelId;      /* Correlation Id related to this subscription */
    MQLONG    PubPriority;      /* Priority set in publications */
    MQBYTE32  PubAccountingToken; /* Accounting Token set in publications */
    MQCHAR32  PubAppIdentityData; /* Appl Identity Data set in publications */
    MQCHARV   SelectionString;  /* Message selector structure */
    MQLONG    SubLevel;         /* Subscription level */
    MQCHARV   ResObjectString;  /* Resolved Long object name*/
    /* Ver:1 */
};
```

### Declaração COBOL para MQSD

```
** Address of variable length string
   20 MQSD-OBJECTSTRING-VSPTR          POINTER.
** Offset of variable length string
```

```

20 MQSD-OBJECTSTRING-VSOFFSET PIC S9(9) BINARY.
** size of buffer
20 MQSD-OBJECTSTRING-VSBUFSIZE PIC S9(9) BINARY.
** Length of variable length string
20 MQSD-OBJECTSTRING-VSLENGTH PIC S9(9) BINARY.
** CCSID of variable length string
20 MQSD-OBJECTSTRING-VSCCSID PIC S9(9) BINARY.
** Subscription name
15 MQSD-SUBNAME.
** Address of variable length string
20 MQSD-SUBNAME-VSPTR POINTER.
** Offset of variable length string
20 MQSD-SUBNAME-VSOFFSET PIC S9(9) BINARY.
** size of buffer
20 MQSD-SUBNAME-VSBUFSIZE PIC S9(9) BINARY.
** Length of variable length string
20 MQSD-SUBNAME-VSLENGTH PIC S9(9) BINARY.
** CCSID of variable length string
20 MQSD-SUBNAME-VSCCSID PIC S9(9) BINARY.
** Subscription User data
15 MQSD-SUBUSERDATA.
** Address of variable length string
20 MQSD-SUBUSERDATA-VSPTR POINTER.
** Offset of variable length string
20 MQSD-SUBUSERDATA-VSOFFSET PIC S9(9) BINARY.
** size of buffer
20 MQSD-SUBUSERDATA-VSBUFSIZE PIC S9(9) BINARY.
** Length of variable length string
20 MQSD-SUBUSERDATA-VSLENGTH PIC S9(9) BINARY.
** CCSID of variable length string
20 MQSD-SUBUSERDATA-VSCCSID PIC S9(9) BINARY.
** Correlation Id related to this subscription
15 MQSD-SUBCORRELID PIC X(24).
** Priority set in publications
15 MQSD-PUBPRIORITY PIC S9(9) BINARY.
** Accounting Token set in publications
15 MQSD-PUBACCOUNTINGTOKEN PIC X(32).
** Appl Identity Data set in publications
15 MQSD-PUBAPPLIDENTITYDATA PIC X(32).
** Message Selector
15 MQSD-SELECTIONSTRING.
** Address of variable length string
20 MQSD-SELECTIONSTRING-VSPTR POINTER.
** Offset of variable length string
20 MQSD-SELECTIONSTRING-VSOFFSET PIC S9(9) BINARY.
** size of buffer
20 MQSD-SELECTIONSTRING-VSBUFSIZE PIC S9(9) BINARY.
** Length of variable length string
20 MQSD-SELECTIONSTRING-VSLENGTH PIC S9(9) BINARY.
** CCSID of variable length string
20 MQSD-SELECTIONSTRING-VSCCSID PIC S9(9) BINARY.
** Selection criteria
20 MQSD-SELECTIONSTRING-SUBLEVEL PIC S9(9) BINARY.
** Long object name
20 MQSD-SELECTIONSTRING-RESOBJSTRING PIC S9(9) BINARY.

```

## Declaração PL/I para MQSD

```

dcl
1 MQSD based,
3 StrucId char(4), /* Structure identifier */
3 Version fixed bin(31), /* Structure version number */
3 Options fixed bin(31), /* Options associated with subscribing */
3 ObjectName char(48), /* Object name */
3 AlternateUserId char(12), /* Alternate user identifier */
3 AlternateSecurityId char(40), /* Alternate security identifier */
3 SubExpiry fixed bin(31), /* Expiry of Subscription */
3 ObjectString, /* Object Long name */
5 VSPtr pointer, /* Address of variable length string */
5 VSOffset fixed bin(31), /* Offset of variable length string */
5 VSBufSize fixed bin(31), /* size of buffer */
5 VSLength fixed bin(31), /* Length of variable length string */
5 VSCCSID fixed bin(31); /* CCSID of variable length string */
3 SubName, /* Subscription name */
5 VSPtr pointer, /* Address of variable length string */
5 VSOffset fixed bin(31), /* Offset of variable length string */
5 VSBufSize fixed bin(31), /* size of buffer */
5 VSLength fixed bin(31), /* Length of variable length string */
5 VSCCSID fixed bin(31); /* CCSID of variable length string */
3 SubUserData, /* Subscription User data */

```

```

5 VSPtr      pointer,      /* Address of variable length string */
5 VSOffset   fixed bin(31), /* Offset of variable length string */
5 VSBufSize  fixed bin(31), /* size of buffer */
5 VSLength   fixed bin(31), /* Length of variable length string */
5 VSCCSID    fixed bin(31), /* CCSID of variable length string */
3 SubCorrelId char(24),    /* Correlation Id related to this subscription */
3 PubPriority fixed bin(31), /* Priority set in publications */
3 PubAccountingToken char(32), /* Accounting Token set in publications */
3 PubApplIdentityData char(32), /* Appl Identity Data set in publications */
3 SelectionString, /* Message Selection */
5 VSPtr      pointer,      /* Address of variable length string */
5 VSOffset   fixed bin(31), /* Offset of variable length string */
5 VSBufSize  fixed bin(31), /* size of buffer */
5 VSLength   fixed bin(31), /* Length of variable length string */
5 VSCCSID    fixed bin(31), /* CCSID of variable length string */
3 SubLevel   fixed bin(31), /* Subscription level */
3 ResObjectString, /* Resolved Long object name */
5 VSPtr      pointer,      /* Address of variable length string */
5 VSOffset   fixed bin(31), /* Offset of variable length string */
5 VSBufSize  fixed bin(31), /* size of buffer */
5 VSLength   fixed bin(31), /* Length of variable length string */
5 VSCCSID    fixed bin(31); /* CCSID of variable length string */

```

## Declaração do High Level Assembler para MQSD

```

MQSD          DSECT
MQSD_STRUCID  DS CL4 Structure identifier
MQSD_VERSION  DS F Structure version number
MQSD-OPTIONS DS F Options associated with subscribing
MQSD_OBJECTNAME DS CL48 Object name
MQSD_ALTERNATEUSERID DS CL12 Alternate user identifier
MQSD_ALTERNATESECURITYID DS CL40 Alternate security identifier
MQSD_SUBEXPIRY DS F Expiry of Subscription
MQSD_OBJECTSTRING DS 0F Object Long name
MQSD_OBJECTSTRING_VSPTR DS F Address of variable length string
MQSD_OBJECTSTRING_VSOFFSET DS F Offset of variable length string
MQSD_OBJECTSTRING_VSBUFSIZE DS F size of buffer
MQSD_OBJECTSTRING_VSLENGTH DS F Length of variable length string
MQSD_OBJECTSTRING_VSCCSID DS F CCSID of variable length string
MQSD_OBJECTSTRING_LENGTH EQU *-MQSD_OBJECTSTRING
ORG MQSD_OBJECTSTRING
MQSD_OBJECTSTRING_AREA DS CL(MQSD_OBJECTSTRING_LENGTH)
*
MQSD_SUBNAME DS 0F Subscription name
MQSD_SUBNAME_VSPTR DS F Address of variable length string
MQSD_SUBNAME_VSOFFSET DS F Offset of variable length string
MQSD_SUBNAME_VSBUFSIZE DS F size of buffer
MQSD_SUBNAME_VSLENGTH DS F Length of variable length string
MQSD_SUBNAME_VSCCSID DS F CCSID of variable length string
MQSD_SUBNAME_LENGTH EQU *-MQSD_SUBNAME
ORG MQSD_SUBNAME
MQSD_SUBNAME_AREA DS CL(MQSD_SUBNAME_LENGTH)
*
MQSD_SUBUSERDATA DS 0F Subscription User data
MQSD_SUBUSERDATA_VSPTR DS F Address of variable length string
MQSD_SUBUSERDATA_VSOFFSET DS F Offset of variable length string
MQSD_SUBUSERDATA_VSBUFSIZE DS F size of buffer
MQSD_SUBUSERDATA_VSLENGTH DS F Length of variable length string
MQSD_SUBUSERDATA_VSCCSID DS F CCSID of variable length string
MQSD_SUBUSERDATA_LENGTH EQU *-MQSD_SUBUSERDATA
ORG MQSD_SUBUSERDATA
MQSD_SUBUSERDATA_AREA DS CL(MQSD_SUBUSERDATA_LENGTH)
*
MQSD_SUBCORRELID DS CL24 Correlation Id related to this subscription
MQSD_PUBPRIORITY DS F Priority set in publications
MQSD_PUBACCOUNTINGTOKEN DS CL32 Accounting Token set in publications
MQSD_PUBAPPLIDENTITYDATA DS CL32 Appl Identity Data set in publications
*
MQSD_SELECTIONSTRING DS F Message Selector
MQSD_SELECTIONSTRING_VSPTR DS F Address of variable length string
MQSD_SELECTIONSTRING_VSOFFSET DS F Offset of variable length string
MQSD_SELECTIONSTRING_VSBUFSIZE DS F size of buffer
MQSD_SELECTIONSTRING_VSLENGTH DS F Length of variable length string
MQSD_SELECTIONSTRING_VSCCSID DS F CCSID of variable length string
MQSD_SELECTIONSTRING_LENGTH EQU *- MQSD_SELECTIONSTRING
ORG MQSD_SELECTIONSTRING
MQSD_SELECTIONSTRING_AREA DS CL(MQSD_SELECTIONSTRING_LENGTH)
*
MQSD-SUBLEVEL DS F Subscription level
*

```

```

MQSD_RESOBJECTSTRING      DS  F   Resolved Long object name
MQSD_RESOBJECTSTRING_VSPTR DS  F   Address of variable length string
MQSD_RESOBJECTSTRING_VSOFFSET DS F   Offset of variable length string
MQSD_RESOBJECTSTRING_VSBUFSIZE DS F   size of buffer
MQSD_RESOBJECTSTRING_VSLENGTH DS F   Length of variable length string
MQSD_RESOBJECTSTRING_VSCCSID DS  F   CCSID of variable length string
MQSD_RESOBJECTSTRING_LENGTH EQU *- MQSD_RESOBJECTSTRING
ORG MQSD_RESOBJECTSTRING
MQSD_RESOBJECTSTRING_AREA DS  CL(MQSD_RESOBJECTSTRING_LENGTH)
*
MQSD_LENGTH              EQU  *-MQSD
ORG  MQSD
MQSD_AREA                DS   CL(MQSD_LENGTH)

```

### **StrucId (MQCHAR4) para MQSD**

Este é o identificador da estrutura do descritor de assinatura. Ele é sempre um campo de entrada Seu valor é MQSD\_STRUC\_ID.

O valor deve ser:.

#### **MQSD\_STRUC\_ID**

Identificador para a estrutura do descritor de assinatura.

Para a linguagem de programação C, a constante MQSD\_STRUC\_ID\_ARRAY também é definida. Ele tem o mesmo valor que MQSD\_STRUC\_ID, mas é uma matriz de caracteres em vez de uma cadeia.

### **Versão (MQLONG) para MQSD**

Este é o número da versão da estrutura; o valor deve ser:

#### **MQSD\_VERSION\_1**

Version-1 Estrutura do descritor de assinatura.

A constante a seguir especifica o número da versão atual:

#### **MQSD\_CURRENT\_VERSION**

Versão atual da estrutura do Descritor de Subscrição

Isto é sempre um campo de entrada. O valor inicial desse campo é MQSD\_VERSION\_1.

### **Opções (MQLONG) para MQSD**

Isso fornece opções para controlar a ação da chamada MQSUB..

Você deve especificar pelo menos uma das opções a seguir:

- MQSO\_ALTER
- MQSO\_RESUME
- MQSO\_CREATE

Para especificar mais de uma opção, inclua os valores juntos (não inclua a mesma constante mais de uma vez) ou combine os valores usando a operação OR bit a bit (se a linguagem de programação suportar operações de bit).

As combinações que não são válidas são observadas neste tópico; quaisquer outras combinações são válidas

**Opções de acesso ou criação:** as opções de acesso e criação controlam se uma assinatura é criada ou se uma assinatura existente é retornada ou alterada. Deve-se especificar pelo menos uma dessas opções

<i>Tabela 526. Combinações válidas de opções de acesso e criação</i>	
<b>Combinação de opções</b>	<b>Notas</b>
MQSO_CREATE	Cria uma assinatura se uma não existir. Essa combinação falha se a assinatura existir.

Tabela 526. Combinações válidas de opções de acesso e criação (continuação)

Combinação de opções	Notas
MQSO_RESUME	Continua uma assinatura existente.. Esta combinação falhará se não existir nenhuma assinatura.
MQSO_CREATE + MQSO_RESUME	Cria uma assinatura se uma não existir e continua uma correspondente, se existir. Essa combinação é útil quando usada em um aplicativo que é executado várias vezes.
MQSO_ALTER (ver nota)	Continua uma assinatura existente, alterando quaisquer campos para corresponder ao especificado no MQSD. Esta combinação falhará se não existir nenhuma assinatura.
MQSO_CREATE + MQSO_ALTER (ver nota)	Cria uma assinatura se uma não existir e retoma uma correspondente, se ela existir, alterando quaisquer campos para corresponder aos especificados no MQSD. Essa combinação é útil quando usada em um aplicativo que deseja assegurar que sua assinatura esteja em um determinado estado antes de continuar.

**Nota:**

Opções que especificam MQSO\_ALTER também podem especificar MQSO\_RESUME, mas essa combinação não tem efeito adicional para especificar MQSO\_ALTER sozinho. MQSO\_ALTER implica MQSO\_RESUME, porque chamar MQSUB para alterar uma assinatura implica que a assinatura também será continuada O oposto não é verdadeiro, no entanto: retomar uma assinatura não implica que ela deve ser alterada.

**MQSO\_CREATE**

Crie uma nova assinatura para o tópico especificado.. Se uma assinatura usando o mesmo *SubName* existir, a chamada falhará com MQRC\_SUB\_ALREADY\_EXISTS. Essa falha pode ser evitada combinando a opção MQSO\_CREATE com MQSO\_RESUME.. O *SubName* nem sempre é necessário Para obter mais detalhes, consulte a descrição desse campo.

Combinar MQSO\_CREATE com MQSO\_RESUME retorna um identificador para uma assinatura pré-existente para o *SubName* especificado se um for localizado; se não houver uma assinatura existente, um novo será criado usando todos os campos fornecidos no MQSD.

MQSO\_CREATE também pode ser combinado com MQSO\_ALTER para efeito semelhante.

**MQSO\_RESUME**

Retorne um identificador para uma assinatura pré-existente que corresponda à especificada por *SubName* Nenhuma mudança é feita nos atributos de assinaturas correspondentes e são retornados na saída na estrutura MQSD. Apenas os seguintes campos MQSD são usados: StrucId, Versão, Opções, AlternateUserId e AlternateSecurityId e SubName.

A chamada falha com o código de razão MQRC\_NO\_SUBSCRIPTION se uma assinatura não existir correspondente ao nome completo da assinatura. Essa falha pode ser evitada combinando a opção MQSO\_CREATE com MQSO\_RESUME..

O ID do usuário da assinatura é o ID do usuário que criou a assinatura ou, se tiver sido alterado posteriormente por um ID do usuário diferente, é o ID do usuário da alteração bem-sucedida mais recente. Se um ID AlternateUserfor usado e o uso de IDs de usuário alternativos for permitido para esse usuário, o ID do usuário alternativo será registrado como o ID do usuário que criou a assinatura, em vez do ID do usuário sob o qual a assinatura foi feita..

Se existir uma assinatura correspondente que foi criada sem a opção MQSO\_ANY\_USERID e o ID do usuário da assinatura for diferente daquele do aplicativo solicitando uma manipulação para a assinatura, a chamada falhará com o código de razão MQRC\_IDENTITY\_MISMATCH.

Se uma assinatura correspondente existir e estiver atualmente em uso, a chamada falhará com MQRC\_SUBSCRIPTION\_IN\_USE.

Se a assinatura nomeada em SubName não for uma assinatura válida para continuar ou alterar a partir de um aplicativo, a chamada falhará com MQRC\_INVALID\_SUBSCRIPTION.

MQSO\_RESUME está implícito por MQSO\_ALTER, portanto, não é necessário combiná-la a essa opção. No entanto, combinar as duas opções não causa erro.

## MQSO\_ALTER

Retorne uma manipulação para uma assinatura pré-existente com o nome completo da assinatura correspondente que foi especificado pelo nome em *SubName*. Quaisquer atributos da assinatura que sejam diferentes daqueles especificados no MQSD são alterados na assinatura, a menos que a alteração seja desaprovada para esse atributo. Os detalhes são anotados na descrição de cada atributo e são resumidos na tabela a seguir: Se você tentar alterar um atributo que não pode ser alterado ou alterar uma assinatura que tenha configurado a opção MQSO\_IMMUTABLE, a chamada falhará com o código de razão mostrado na tabela a seguir.

A chamada falhará com o código de razão MQRC\_NO\_SUBSCRIPTION se uma assinatura correspondente ao nome completo da assinatura não existir. É possível evitar essa falha, combinando a opção MQSO\_CREATE com MQSO\_ALTER.

Combinar MQSO\_CREATE com MQSO\_ALTER retorna uma manipulação para uma assinatura pré-existente para o *SubName* especificado se uma for localizada; se não houver uma assinatura existente, uma nova será criada usando todos os campos fornecidos no MQSD.

O ID do usuário da assinatura é o ID do usuário que criou a assinatura ou, se posteriormente for alterado por um ID do usuário diferente, será o ID do usuário da alteração mais recente e bem-sucedida. Se um ID AlternateUser for usado e o uso de IDs de usuário alternativos for permitido para esse usuário, o ID do usuário alternativo será registrado como o ID do usuário que criou a assinatura, em vez do ID do usuário sob o qual a assinatura foi feita..

Se existir uma assinatura correspondente que foi criada sem a opção MQSO\_ANY\_USERID e o ID do usuário da assinatura for diferente daquele do aplicativo solicitando um identificador para a assinatura, a chamada falhará com o código de razão MQRC\_IDENTITY\_MISMATCH.

Se uma assinatura correspondente existir e estiver atualmente em uso, a chamada falhará com MQRC\_SUBSCRIPTION\_IN\_USE.

Se a assinatura nomeada em SubName não for uma assinatura válida para continuar ou alterar a partir de um aplicativo, a chamada falhará com MQRC\_INVALID\_SUBSCRIPTION.

A tabela a seguir mostra a capacidade de MQSO\_ALTER para alterar os valores de atributos em MQSD e MQSUB.

*Tabela 527. Atributos em MQSD e MQSUB que podem ser alterados*

Descritor de tipo de dados ou chamada de função	Nome de Campo	Esse atributo pode ser alterado usando MQSO_ALTER	Código de razão
MQSD	Opções de durabilidade:	No	MQRC_DURABILITY_NOT_ALTERABLE
MQSD	Opções de Destino	Sim	Nenhum
MQSD	Opções de registro	Sim (consulte a nota "1" na página 596)	MQRC_GROUPING_NOT_ALTERABLE se você tentar alterar MQSO_GROUP_SUB
MQSD	Opções de publicação	Sim (consulte a nota "2" na página 596)	Nenhum
MQSD	Opções de curinga	No	MQRC_TOPIC_NOT_ALTERABLE
MQSD	Outras Opções	Não (ver nota "3" na página 596)	Nenhum

Tabela 527. Atributos em MQSD e MQSUB que podem ser alterados (continuação)

Descritor de tipo de dados ou chamada de função	Nome de Campo	Esse atributo pode ser alterado usando MQSO_ALTER	Código de razão
MQSD	ObjectName	No	MQRC_TOPIC_NOT_ALTERABLE
MQSD	AlternateUserId	Não (ver nota "4" na página 596)	Nenhum
MQSD	AlternateSecurityId	Não (ver nota "4" na página 596)	Nenhum
MQSD	SubExpiry	Sim	Nenhum
MQSD	ObjectString	No	MQRC_TOPIC_NOT_ALTERABLE
MQSD	SubName	Não (ver nota "5" na página 596)	Nenhum
MQSD	SubUserData	Sim	Nenhum
MQSD	SubCorrelId	Sim (consulte a nota "6" na página 596)	MQRC_GROUPING_NOT_ALTERABLE quando em uma assinatura agrupada
MQSD	PubPriority	Sim	Nenhum
MQSD	PubAccountingToken	Sim	Nenhum
MQSD	PubApplIdentityData	Sim	Nenhum
MQSD	SubLevel	No	MQRC_SUBLEVEL_NOT_ALTERABLE
MQSUB	Hobj	Sim (consulte a nota "6" na página 596)	MQRC_GROUPING_NOT_ALTERABLE quando em uma assinatura agrupada

#### Notas:

1. MQSO\_GROUP\_SUB não pode ser alterado.
2. MQSO\_NEW\_PUBLICATIONS\_ONLY não pode ser alterado porque não faz parte da assinatura
3. Essas opções não fazem parte da assinatura
4. Este atributo não faz parte da assinatura
5. Este atributo é a identidade da assinatura sendo alterada
6. Alterável, exceto quando parte de um sub agrupado (MQSO\_GROUP\_SUB)

**Opções de Durabilidade:** As opções a seguir controlam a durabilidade da assinatura. É possível especificar apenas uma destas opções Se estiver alterando uma assinatura existente usando a opção MQSO\_ALTER, não será possível alterar a durabilidade da assinatura. No retorno de uma chamada MQSUB usando MQSO\_RESUME, a opção de durabilidade apropriada é configurada..

#### MQSO\_DURABLE

Solicite que a assinatura para este tópico permaneça até ser explicitamente removida usando MQCLOSE com a opção MQCO\_REMOVE\_SUB. Se essa assinatura não for explicitamente removida, ela permanecerá mesmo depois que essa conexão de aplicativos com o gerenciador de filas for encerrada

Se uma assinatura durável for solicitada para um tópico definido como não permitindo assinaturas duráveis, a chamada falhará com MQRC\_DURABILITY\_NOT\_ALLOWED.

#### MQSO\_NON\_DURABLE

Solicite que a assinatura desse tópico seja removida quando a conexão de aplicativos com o gerenciador de filas for encerrada, se ela ainda não tiver sido explicitamente removida explicitamente. MQSO\_NON\_DURABLE é o oposto da opção MQSO\_DURABLE e é definido para a documentação do programa de auxílio Ele será o padrão se nenhum for especificado

**Opções de destino:** a opção a seguir controla o destino para o qual as publicações de um tópico que foi inscrito são enviadas.. Se alterar uma assinatura existente usando a opção MQSO\_ALTER, o destino usado para publicações para a assinatura poderá ser alterado. No retorno de uma chamada MQSUB usando MQSO\_RESUME, essa opção é configurada, se apropriado,



## **MQSO\_MANAGED**

Solicite que o destino para o qual as publicações são enviadas seja gerenciado pelo gerenciador de filas

A manipulação de objetos retornada em *Hobj* representa uma fila gerenciada pelo gerenciador de filas e é para uso com as chamadas MQGET, MQCB, MQINQ ou MQCLOSE subsequentes

Uma manipulação de objetos retornada de uma chamada MQSUB anterior não pode ser fornecida no parâmetro **Hobj** quando MQSO\_MANAGED não é especificado.

## **MQSO\_NO\_MULTICAST**

Solicite que o destino ao qual as publicações são enviadas não seja um endereço de grupo multicast. Esta opção é válida apenas quando combinada com a opção MQSO\_MANAGED Quando um identificador para uma fila é fornecido no parâmetro **Hobj** , o multicast não pode ser usado para essa assinatura e a opção não é válida

Se o tópico for definido para permitir apenas assinaturas de multicast, usando a configuração MCAST (ONLY) , a chamada falhará com o código de razão MQRC\_MULTICAST\_REQUIRED

**Opção de Escopo:** A opção a seguir controla o escopo da assinatura que está sendo feita. Se alterar uma assinatura existente usando a opção MQSO\_ALTER, essa opção de escopo de assinatura não poderá ser alterada. Ao retornar de uma chamada MQSUB usando MQSO-RESUME, a opção de escopo apropriada é configurada.

## **MQSO\_SCOPE\_QMGR**

Esta assinatura é feita apenas no gerenciador de filas locais Nenhuma assinatura de proxy é distribuída para outros gerenciadores de filas na rede Apenas as publicações que são publicadas neste gerenciador de filas são enviadas para este assinante Isso substitui qualquer conjunto de comportamento usando o atributo de tópico SUBSCOPE

**Nota:** Se não configurado, o escopo da subscrição será determinado pelo atributo do tópico SUBSCOPE

**Opções de registro:** As opções a seguir controlam os detalhes do registro que é feito para o gerenciador de filas para essa assinatura. Se alterar uma assinatura existente usando a opção MQSO\_ALTER, essas opções de registro poderão ser mudadas. No retorno de uma chamada MQSUB usando MQSO-RESUME, as opções de registro apropriadas são configuradas.

## **MQSO\_GROUP\_SUB**

Essa assinatura deve ser agrupada com outras assinaturas do mesmo SubLevel usando a mesma fila e especificando o mesmo ID de correlação para que quaisquer publicações para tópicos que causariam mais de uma mensagem de publicação a ser fornecida para o grupo de assinaturas, devido a um conjunto de sobreposições de sequências de tópicos que está sendo usado, apenas uma mensagem a ser entregue para a fila. Se essa opção não for usada, cada assinatura exclusiva (identificada por SubName) que corresponde será fornecida com uma cópia da publicação, o que poderia significar que mais de uma cópia da publicação pode ser colocada na fila compartilhada por um número de assinaturas.

Apenas a assinatura mais significativa do grupo é fornecida com uma cópia da publicação. A assinatura mais significativa é baseada no nome do tópico Completo até o ponto em que um curinga é localizado Se uma mistura de esquemas curingas for usada dentro do grupo, apenas a posição do curinga será importante Você é aconselhado a não combinar diferentes esquemas curinga em um grupo de assinaturas que compartilham a mesma fila.

Ao criar uma nova assinatura agrupada, ela ainda deverá ter um SubName exclusivo, mas se corresponder ao nome completo do tópico de uma assinatura existente no grupo, a chamada falhará com MQRC\_DUPLICATE\_GROUP\_SUB.

Se a assinatura mais significativa no grupo também especificar MQSO\_NOT\_OWN\_PUBS e esta for uma publicação do mesmo aplicativo, nenhuma publicação será entregue para a fila

Ao alterar uma assinatura feita com esta opção, os campos que implicam o agrupamento, Hobj na chamada MQSUB (representando a fila e o nome do gerenciador de filas) e o

ID SubCorrelnão podem ser alterados. Tentar alterá-los faz com que a chamada falhe com MQRG\_GROUPING\_NOT\_ALTERABLE.

Essa opção deve ser combinada com MQSO\_SET\_CORREL\_ID com um ID SubCorrelque não está configurado como MQCI\_NONE e não pode ser combinado com MQSO\_MANAGED.

### **MQSO\_ANY\_USERID**

Quando MQSO\_ANY\_USERID é especificado, a identidade do assinante não é restrita a um único ID do usuário. Isso permite que qualquer usuário altere ou continue a assinatura quando tem autoridade adequada. Apenas um único usuário pode ter a assinatura a qualquer momento. Uma tentativa de continuar o uso de uma assinatura atualmente em uso por outro aplicativo faz a chamada falhar com MQRG\_SUBSCRIPTION\_IN\_USE.

Para incluir essa opção em uma assinatura existente, a chamada MQSUB (usando MQSO\_ALTER) deve ser proveniente do mesmo ID do usuário que a própria assinatura original.

Se uma chamada MQSUB se referir a uma assinatura existente com MQSO\_ANY\_USERID configurado e o ID do usuário for diferente da assinatura original, a chamada será bem-sucedida apenas se o novo ID do usuário tiver autoridade para assinar o tópico. Na conclusão bem-sucedida, publicações futuras para este assinante são colocadas na fila de assinantes com o novo ID do usuário configurado na mensagem de publicação.

Não especifique MQSO\_ANY\_USERID e MQSO\_FIXED\_USERID. Se nenhum for especificado, o padrão será MQSO\_FIXED\_USERID.

### **MQSO\_FIXED\_USERID**

Quando MQSO\_FIXED\_USERID é especificado, a assinatura pode ser alterada ou retomada apenas pelo último ID do usuário para alterar a assinatura. Se a assinatura não tiver sido alterada, será o ID do usuário que criou a assinatura.

Se um verbo MQSUB referir-se a uma assinatura existente com MQSO\_ANY\_USERID configurado e alterar a assinatura usando MQSO\_ALTER para usar a opção MQSO\_FIXED\_USERID, o ID do usuário da assinatura agora será fixado nesse novo ID do usuário. A chamada será bem-sucedida apenas se o novo ID do usuário tiver autoridade para assinar o tópico.

Se um ID do usuário diferente daquele registrado como proprietário de uma assinatura tentar continuar ou alterar uma assinatura MQSO\_FIXED\_USERID, a chamada falhará com MQRG\_IDENTITY\_MISMATCH. O ID do usuário proprietário de uma assinatura pode ser visualizado usando o comando DISPLAY SBSTATUS.

Não especifique MQSO\_ANY\_USERID e MQSO\_FIXED\_USERID. Se nenhum for especificado, o padrão será MQSO\_FIXED\_USERID.

**Opções de publicação:** As opções a seguir controlam a maneira como as publicações são enviadas para este assinante... Se alterar uma assinatura existente usando a opção MQSO\_ALTER, essas opções de publicação poderão ser alteradas.

### **MQSO\_NOT\_OWN\_PUBS**

Informa ao broker que o aplicativo não deseja ver nenhuma de sua própria publicação. As publicações são consideradas originadas do mesmo aplicativo se os identificadores de conexão forem os mesmos. No retorno de uma chamada MQSUB usando MQSO\_RESUME, essa opção é configurada, se apropriado,

### **MQSO\_NEW\_PUBLICATIONS\_ONLY**

Nenhuma publicação retida no momento deve ser enviada, quando essa assinatura for criada, apenas novas publicações; Essa opção se aplica apenas quando MQSO\_CREATE é especificado.. Quaisquer mudanças subsequentes em uma assinatura não alteram o fluxo de publicações e, portanto, quaisquer publicações retidas em um tópico já terão sido enviadas ao assinante como novas publicações.

Se essa opção for especificada sem MQSO\_CREATE, a chamada falhará com MQRG\_OPTIONS\_ERROR. No retorno de uma chamada MQSUB usando MQSO\_RESUME, essa opção não será configurada mesmo se a assinatura foi criada usando essa opção.

Se essa opção não for usada, as mensagens retidas anteriormente serão enviadas para a fila de destino fornecida. Se essa ação falhar devido a um erro, MQRC\_RETAINED\_MSG\_Q\_ERROR ou MQRC\_RETAINED\_NOT\_ENTREGUE, a criação da assinatura falhará.

### **MQSO\_PUBLICATIONS\_ON\_REQUEST**

Configurar essa opção indica que o assinante solicitará informações especificamente quando necessário. O gerenciador de filas não envia mensagens não solicitadas ao assinante. A publicação retida (ou possivelmente várias publicações se um curinga for especificado no tópico) é enviada ao assinante cada vez que uma chamada MQSUBRQ é feita usando o identificador Hsub de uma chamada MQSUB anterior. Nenhuma publicação é enviada como um resultado da chamada MQSUB usando esta opção. No retorno de uma chamada MQSUB usando MQSO\_RESUME, essa opção é configurada, se apropriado,

Essa opção não é válida em combinação com um SubLevel maior que 1.

**Opções de leitura antecipada:** as opções a seguir controlam se mensagens não persistentes são enviadas para um aplicativo antes do aplicativo que as solicita.

### **MQSO\_READ\_AHEAD\_AS\_Q\_DEF**

Se a chamada MQSUB usar um identificador gerenciado, o atributo de leitura antecipada padrão da fila modelo associada ao tópico inscrito para determinar se as mensagens são enviadas para o aplicativo antes que o aplicativo as solicite.

Esse é o valor-padrão.

### **MQSO\_NO\_READ\_AHEAD**

Se a chamada MQSUB usar um identificador gerenciado, as mensagens não serão enviadas ao aplicativo antes que o aplicativo as solicite.

### **MQSO\_READ\_AHEAD**

Se a chamada MQSUB usar um identificador gerenciado, as mensagens poderão ser enviadas ao aplicativo antes que o aplicativo as solicite.

### **Nota:**

As notas a seguir se aplicam às opções de leitura antecipada:

1. Apenas uma dessas opções pode ser especificada.. Se MQSO\_READ\_AHEAD e MQSO\_NO\_READ\_AHEAD forem especificados, o código de razão MQRC\_OPTIONS\_ERROR será retornado. Essas opções são aplicáveis apenas se MQSO\_MANAGED for especificado.
2. Eles não são aplicáveis para MQSUB quando uma fila é transmitida que foi aberta anteriormente... A leitura antecipada pode não ser ativada quando solicitada. As opções MQGET usadas na primeira chamada MQGET podem evitar que a leitura antecipada seja ativada. Além disso, a leitura antecipada é desativada quando o cliente está se conectando a um gerenciador de filas no qual a leitura antecipada não é suportada. Se o aplicativo não estiver executando como um cliente IBM MQ, essas opções serão ignoradas.

**Opções de curinga:** As opções a seguir controlam como os curingas são interpretados na sequência fornecida no campo ObjectString do MQSD.. É possível especificar apenas uma destas opções. Se alterar uma assinatura existente usando a opção MQSO\_ALTER, essas opções curingas não poderão ser alteradas. No retorno de uma chamada MQSUB usando MQSO\_RESUME, a opção curinga apropriada é configurada.

### **MQSO\_WILDCARD\_CHAR**

Curingas operam apenas em caracteres dentro da cadeia de tópicos.

O comportamento definido por MQSO\_WILDCARD\_CHAR é mostrado na tabela a seguir:

<i>Tabela 528. Como curingas são interpretados</i>	
<b>Caractere especial</b>	<b>Comportamento</b>
Barra (/)	Sem significado, apenas outro personagem

<i>Tabela 528. Como curingas são interpretados (continuação)</i>	
<b>Caractere especial</b>	<b>Comportamento</b>
Asterisco (*)	Curinga, zero ou mais caracteres
Ponto de interrogação (?)	Curinga, 1 caractere
Sinal de percentual (%)	Caractere de escape para permitir que os caracteres (*), (?) ou (%) sejam usados em uma sequência e não sejam interpretados como um caractere especial, por exemplo, (% *), (%?) ou (%%)

Por exemplo, publicar no tópico a seguir:

```
/level0/level1/level2/level3/level4
```

corresponde assinantes usando os tópicos a seguir:

```
*
/*
/ level0/level1/level2/level3/*
/ level0/level1/*/level3/level4
/ level0/level1/level2/level3/level4
```

**Nota:** Esse uso de curingas fornece exatamente o significado fornecido em IBM MQ V6 e WebSphere MB V6 ao usar mensagens formatadas MQRFH1 para publicação / assinatura. É recomendado que isso não seja usado para aplicativos recém-gravados e seja usado apenas para aplicativos que estavam em execução anteriormente com relação a essa versão e não foram alterados para usar o comportamento de curinga padrão, conforme descrito em MQSO\_WILDCARD\_TOPIC.

### **MQSO\_WILDCARD\_TOPIC**

Os curingas operam apenas em elementos de tópico dentro da sequência de tópicos Este é o comportamento padrão se nenhum for escolhido.

O comportamento requerido por MQSO\_WILDCARD\_TOPIC é mostrado na seguinte tabela:

<i>Tabela 529. Como curingas são interpretados</i>	
<b>Caractere especial</b>	<b>Comportamento</b>
(/)	Separador de nível de tópico..
sinal de tralha (#)	Curinga: nível de tópico múltiplo
Símbolo de mais (+)	Curinga: nível de tópico único

#### **Notas:**

O (+) e (#) não são tratados como curingas se eles forem combinados com outros caracteres (incluindo eles mesmos) dentro de um nível de tópico. Na sequência a seguir, os caracteres (#) e (+) são tratados como caracteres comuns.

```
level0/level1/#+/level3/level#
```

Por exemplo, publicar no tópico a seguir:

```
/level0/level1/level2/level3/level4
```

corresponde assinantes usando os tópicos a seguir:

```
#
/#
/ level0/level1/level2/level3/#
/ level0/level1+/level3/level4
```

**Outras opções:** as opções a seguir controlam a maneira como a chamada API é emitida em vez da assinatura. No retorno de uma chamada do MQSUB usando MQSO\_RESUME, essas opções não são alteradas. Consulte [“ID do AlternateUser\(MQCHAR12\) para MQSD”](#) na página 602 para obter mais detalhes.

### MQSO\_ALTERNATE\_USER\_AUTHORITY

O campo `AlternateUserId` contém um identificador de usuário a ser usado para validar esta chamada MQSUB. A chamada poderá ser bem-sucedida somente se esse ID `AlternateUser` estiver autorizado a abrir o objeto com as opções de acesso especificadas, independentemente de o identificador de usuário sob o qual o aplicativo está em execução estar autorizado a fazê-lo.

### MQSO\_SET\_CORREL\_ID

A assinatura é para usar o identificador de correlação fornecido no campo `SubCorrelId`. Se essa opção não for especificada, um identificador de correlação será criado automaticamente pelo gerenciador de fila no momento da assinatura e será retornado para o aplicativo no campo `SubCorrelId`. Para obter mais informações, consulte [“SubCorrelId \(MQBYTE24\) para MQSD”](#) na página 605 para obter mais informações.

Esta opção não pode ser combinada com MQSO\_MANAGED.

### MQSO\_SET\_IDENTITY\_CONTEXT

A assinatura é para usar o token de contabilidade e dados de identidade do aplicativo fornecidos nos campos `PubAccountingToken` e `PubApplIdentityData`.

Se esta opção for especificada, a mesma verificação de autorização será executada como se a fila de destino tivesse sido acessada usando uma chamada MQOPEN com MQOO\_SET\_IDENTITY\_CONTEXT, exceto no caso em que a opção MQSO\_MANAGED também é usada; neste caso, não há nenhuma verificação de autorização na fila de destino.

Se esta opção não for especificada, as publicações enviadas a este assinante terão informações de contexto padrão associadas a elas da seguinte maneira:

<i>Tabela 530. Informações de contexto padrão para publicações enviadas para este assinante</i>	
<b>Campo no MQMD</b>	<b>Valor Usado</b>
<i>UserIdentifier</i>	O ID do usuário associado à assinatura no momento em que ela foi feita.
<i>AccountingToken</i>	Determinado a partir do ambiente, se possível; configurado como MQACT_NONE, se não.
<i>ApplIdentityData</i>	Configurar para espaços em branco..

Esta opção é válida apenas com MQSO\_CREATE e MQSO\_ALTER. Se usado com MQSO\_RESUME, os campos `PubAccountingToken` e `PubApplIdentityData` são ignorados, portanto, essa opção não tem efeito.

Se uma assinatura for alterada sem usar essa opção onde anteriormente as informações de contexto de identidade fornecidas pela assinatura, as informações de contexto padrão serão geradas para a assinatura alterada.

Se uma assinatura permitindo que diferentes IDs de usuário a usem com a opção MQSO\_ANY\_USERID for continuada por um ID do usuário diferente, o contexto de identidade padrão será gerado para o novo ID do usuário que agora possui a assinatura e todas as publicações subsequentes serão entregues contendo o novo contexto de identidade.

## **MQSO\_FAIL\_IF QUIESCING**

A chamada MQSUB falhará se o gerenciador de filas estiver no estado de quiesce No z/OS, para um aplicativo CICS ou IMS , essa opção também força a chamada MQSUB a falhar se a conexão estiver no estado de quiesce

### ***ObjectName (MQCHAR48) para MQSD***

Este é o nome do objeto de tópico conforme definido na gerenciador de filas locais.

O nome pode conter os seguintes caracteres:

- Caracteres alfabéticos maiúsculos (A a Z)
- Caracteres alfabéticos minúsculos (a a z)
- Dígitos numéricos (0 a 9)
- Ponto (.), barra (/), sublinhado (\_), porcentagem (%)

O nome não deve conter espaços em branco iniciais ou integrados, mas pode conter rastreios de espaço em branco. Use um caractere nulo para indicar o término de dados significativos no nome; o nulo e quaisquer caracteres que o seguem são tratados como espaços em branco. As restrições a seguir aplicam-se aos ambientes indicados:

- Em sistemas que usam EBCDIC Katakana, caracteres minúsculos não podem ser usados.
- No z/OS:
  - Evite nomes que começam ou terminam com um sublinhado; eles não podem ser processados pelas operações e painéis de controle.
  - O caractere de porcentagem tem um significado especial para RACF Se RACF for usado como o gerenciador de segurança externa, os nomes não deverão conter o percentual Se forem, esses nomes não serão incluídos em nenhuma verificação de segurança quando os perfis genéricos do RACF forem usados.
- No IBM i, os nomes contendo caracteres minúsculos, barra ou porcentagem devem ser colocados entre aspas quando especificados em comandos. Essas aspas não devem ser especificadas para nomes que ocorrem como campos em estruturas ou como parâmetros em chamadas..

O *ObjectName* é usado para formar o nome completo do tópico

O nome completo do tópico pode ser construído de dois campos diferentes: *ObjectName* e *ObjectString*. Para obter detalhes de como esses dois campos são utilizados, consulte [Combinação de sequências de tópicos](#)

Se o objeto identificado pelo campo *ObjectName* não puder ser localizado, a chamada falhará com o código de razão MQRC\_UNKNOWN\_OBJECT\_NAME, mesmo se houver uma cadeia especificada em *ObjectString*

No retorno de uma chamada MQSUB usando a opção MQSO\_RESUME, esse campo fica inalterado

O comprimento desse campo é fornecido por MQ\_TOPIC\_NAME\_LENGTH O valor inicial deste campo é a cadeia nula em C e 48 caracteres em branco em outras linguagens de programação.

Se alterar uma assinatura existente usando a opção MQSO\_ALTER, o nome do objeto do tópico inscrito não poderá ser alterado. Este campo e o campo *ObjectString* podem ter sido omitidos Se eles forem fornecidos, eles deverão ser resolvidos para o mesmo nome do tópico completo Caso contrário, a chamada falhará com MQRC\_TOPIC\_NOT\_ALTERABLE.

### ***ID do AlternateUser(MQCHAR12) para MQSD***

Se você especificar MQSO\_ALTERNATE\_USER\_AUTHORITY, esse campo conterá um identificador de usuário alternativo que é usado para verificar a autorização para a assinatura e para a saída para a fila de

destino (especificada no parâmetro **Hobj** da chamada MQSUB), no lugar do identificador de usuário sob o qual o aplicativo está atualmente em execução.

Se bem-sucedido, o identificador de usuário especificado nesse campo será registrado como o identificador de usuário proprietário da assinatura no lugar do identificador de usuário sob o qual o aplicativo está atualmente em execução.

Se MQSO\_ALTERNATE\_USER\_AUTHORITY for especificado e este campo estiver totalmente em branco até o primeiro caractere nulo ou o final do campo, a assinatura poderá ser bem-sucedida apenas se nenhuma autorização do usuário for necessária para assinar este tópico com as opções especificadas ou a fila de destino para saída..

Se MQSO\_ALTERNATE\_USER\_AUTHORITY não for especificado, esse campo será ignorado..

As seguintes diferenças existem nos ambientes indicados:

- No z/OS, somente os primeiros 8 caracteres do ID AlternateUser são usados para verificar a autorização para a assinatura. No entanto, o identificador de usuário atual deve ser autorizado a especificar esse identificador de usuário alternativo específico; todos os 12 caracteres do identificador de usuário alternativo são usados para essa verificação... O identificador de usuário deve conter apenas caracteres permitidos pelo gerenciador de segurança externo..

No retorno de uma chamada MQSUB usando MQSO\_RESUME, esse campo permanece inalterado

Este é um campo de entrada. O comprimento desse campo é fornecido por MQ\_USER\_ID\_LENGTH O valor inicial deste campo é a sequência nula em C e 12 caracteres em branco em outras linguagens de programação.

### ***AlternateSecurityId (MQBYTE40) para MQSD***

Este é um identificador de segurança que é transmitido com o AlternateUserId para o serviço de autorização para permitir que verificações de autorização apropriadas sejam executadas.

AlternateSecurityId é usado somente se MQSO\_ALTERNATE\_USER\_AUTHORITY for especificado e o campo AlternateUserId não estiver completamente em branco até o primeiro caractere nulo ou no final do campo.

No retorno de uma chamada MQSUB usando MQSO\_RESUME, esse campo permanece inalterado

Consulte a descrição de [“AlternateSecurityId \(MQBYTE40\) para MQOD”](#) na página 508 no tipo de dados MQOD para obter mais informações..

### ***SubExpiry (MQLONG) para MQSD***

Este é o tempo expresso em décimos de segundo após o qual a assinatura expira. Nenhuma outra publicação corresponderá a esta subscrição após este intervalo ter passado Assim que uma assinatura expira, as publicações não são mais enviadas para a fila No entanto, as publicações que já existem não são afetadas de forma alguma. *SubExpiry* não tem efeito sobre a expiração da publicação

O valor especial a seguir é reconhecido:

#### **MQEI\_UNLIMITED**

A assinatura tem um prazo de expiração ilimitado..

Se alterar uma assinatura existente usando a opção MQSO\_ALTER, a expiração da assinatura poderá ser mudada.

No retorno de uma chamada MQSUB usando a opção MQSO\_RESUME, esse campo é configurado para a expiração original da assinatura e não para o tempo de expiração restante

### ***ObjectString (MQCHARV) para MQSD***

Este é o nome do objeto longo a ser usado

O *ObjectString* é usado para formar o nome do tópico Completo

O nome completo do tópico pode ser construído de dois campos diferentes: *ObjectName* e *ObjectString*. Para obter detalhes de como esses dois campos são utilizados, consulte [Combinação de sequências de tópicos](#)

O comprimento máximo de *ObjectString* é 10240.

Se *ObjectString* não for especificado corretamente, de acordo com a descrição de como usar a estrutura *MQCHARV* ou se exceder o comprimento máximo, a chamada falhará com o código de razão *MQRC\_OBJECT\_STRING\_ERROR*.

Este é um campo de entrada. Os valores iniciais dos campos nessa estrutura são os mesmos da estrutura *MQCHARV*.

Se houver curingas no *ObjectString*, a interpretação desses curingas poderá ser controlada usando as opções Curinga especificadas no campo Opções do *MQSD*.

No retorno de uma chamada *MQSUB* usando a opção *MQSO\_RESUME*, esse campo fica inalterado O nome completo do tópico usado será retornado no campo *ResObjectString*, se um buffer for fornecido

Se alterar uma assinatura existente usando a opção *MQSO\_ALTER*, o nome longo do objeto do tópico inscrito não poderá ser alterado. Este campo e o campo *ObjectName* podem ter sido omitidos Se eles forem fornecidos, eles deverão ser resolvidos para o mesmo nome de tópico completo ou a chamada falhará com *MQRC\_TOPIC\_NOT\_ALTERABLE*

### ***SubName (MQCHARV) para MQSD***

Isso especifica o nome da assinatura Esse campo será necessário apenas se o *Options* especificar a opção *MQSO\_DURABLE*, mas se fornecido, será usado pelo gerenciador de filas para *MQSO\_NON\_DURABLE* também

Se especificado, *SubName* deve ser exclusivo no gerenciador de filas, porque é o método usado para identificar a assinatura.

O comprimento máximo de *SubName* é 10240.

Este campo tem dois propósitos. Para uma assinatura *MQSO\_DURABLE*, use esse campo para identificar uma assinatura para que seja possível retomá-la após ela ter sido criada se você tiver fechado a manipulação para a assinatura (usando a opção *MQCO\_KEEP\_SUB*) ou tiver sido desconectado do gerenciador de filas. Isso é feito usando a chamada *MQSUB* com a opção *MQSO\_RESUME*. Também é exibido na visualização administrativa de assinaturas no campo *SUBID* em *DISPLAY SBSTATUS*.

Se *SubName* for especificado incorretamente, de acordo com a descrição de como usar a estrutura *MQCHARV*, será deixado fora quando for necessário (ou seja, *SubName.VSLength* é zero) ou, se ele exceder o comprimento máximo, a chamada falhará com o código de razão *MQRC\_SUB\_NAME\_ERROR*.

Este é um campo de entrada. Os valores iniciais dos campos nessa estrutura são os mesmos da estrutura *MQCHARV*.

Se alterar uma assinatura existente usando a opção *MQSO\_ALTER*, o nome da assinatura não poderá ser alterado, porque é o campo de identificação usado para localizar a assinatura referenciada. Ele não é alterado na saída de uma chamada de *MQSUB* com a opção *MQSO\_RESUME*

### ***Dados de SubUser(MQCHARV) para MQSD***

Isso especifica os dados do usuário da subscrição Os dados fornecidos na assinatura nesse campo serão incluídos como a propriedade de mensagem *MQSubUserData* de cada publicação enviada para essa assinatura.

O comprimento máximo de *SubUserData* é 10240.



Se *SubUserData* for especificado incorretamente, de acordo com a descrição de como usar a estrutura *MQCHARV* ou se exceder o comprimento máximo, a chamada falhará com o código de razão *MQRC\_SUB\_USER\_DATA\_ERROR*.

Este é um campo de entrada. Os valores iniciais dos campos nessa estrutura são os mesmos da estrutura *MQCHARV*.

Se alterar uma assinatura existente usando a opção *MQSO\_ALTER*, os dados do usuário da assinatura poderão ser alterados.

Este campo de comprimento variável é retornado na saída de uma chamada *MQSUB* usando a opção *MQSO\_RESUME*, se um buffer for fornecido e houver um comprimento de buffer positivo em *VSBuflen*. Se nenhum buffer for fornecido na chamada, somente o comprimento da data do usuário da assinatura será retornado no campo *VSLength* do *MQCHARV*. Se o buffer fornecido for menor que o espaço necessário para retornar o campo, apenas *VSBuflen* bytes serão retornados no buffer fornecido..

### ***SubCorrelId (MQBYTE24) para MQSD***

Este campo contém um identificador de correlação comum a todas as publicações correspondentes a esta assinatura.



**Atenção:** Um identificador de correlação só pode ser transmitido entre os gerenciadores de filas em um cluster de publicação/assinatura, não uma hierarquia.

Todas as publicações enviadas para corresponder a esta assinatura contêm este identificador de correlação no descritor de mensagem. Se várias assinaturas obtiverem suas publicações da mesma fila, usar *MQGET* por identificador de correlação permitirá que apenas publicações para uma assinatura específica sejam obtidas. Esse identificador de correlação pode ser gerado pelo gerenciador de filas ou pelo usuário.

Se a opção *MQSO\_SET\_CORREL\_ID* não for especificada, o identificador de correlação será gerado pelo gerenciador de filas e esse campo será um campo de saída contendo o identificador de correlação que será configurado em cada mensagem publicada para essa assinatura. O identificador de correlação gerado consiste em um identificador do produto de 4 bytes (*AMQX* ou *CSQM* em *ASCII* ou *EBCDIC*) seguido por uma implementação específica do produto de uma sequência exclusiva.

Se a opção *MQSO\_SET\_CORREL\_ID* for especificada, o identificador de correlação será gerado pelo usuário e este campo será um campo de entrada contendo o identificador de correlação a ser configurado em cada publicação para esta assinatura. Nesse caso, se o campo contiver *MQCI\_NONE*, o identificador de correlação que é configurado em cada mensagem publicada para essa assinatura será o identificador de correlação criado pela colocação original da mensagem.

Se a opção *MQSO\_GROUP\_SUB* for especificada e o identificador de correlação especificado for o mesmo que uma assinatura agrupada existente usando a mesma fila e uma sequência de tópicos sobreposta, somente a assinatura mais significativa do grupo será fornecida com uma cópia da publicação.

O comprimento desse campo é fornecido por *MQ\_CORREL\_ID\_LENGTH*. O valor inicial desse campo é *MQCI\_NONE*.

Se você estiver alterando uma assinatura existente usando a opção *MQSO\_ALTER*, e esse campo for um campo de entrada, o identificador de correlação de assinatura poderá ser mudado, a menos que a assinatura seja uma assinatura agrupada, ou seja, ela foi criada usando a opção *MQSO\_GROUP\_SUB*, nesse caso, o identificador de correlação de assinatura não pode ser mudado..

No retorno de uma chamada *MQSUB* usando *MQSO\_RESUME*, esse campo é configurado para o identificador de correlações atual para a assinatura

### ***PubPriority (MQLONG) para MQSD***

Esse é o valor que estará no campo *Priority* do Message Descriptor (*MQMD*) de todas as mensagens de publicação correspondentes a esta assinatura. Para obter mais informações sobre o campo *Priority* no *MQMD*, consulte [“Prioridade \(MQLONG\) para MQMD”](#) na página 467..

O valor deve ser maior ou igual a zero; zero é a prioridade mais baixa. Os valores especiais a seguir também podem ser usados:

### **MQPRI\_PRIORITY\_AS\_Q\_DEF**

Quando uma fila de assinaturas é fornecida no campo *Hobj* na chamada MQSUB e não é um identificador gerenciado, a prioridade para a mensagem é obtida do atributo **DefPriority** dessa fila. Se a fila for uma fila de clusters ou houver mais de uma definição no caminho de resolução do nome da fila, a prioridade será determinada quando a mensagem de publicação for colocada na fila conforme descrito para [“Prioridade \(MQLONG\) para MQMD”](#) na página 467.

Se a chamada MQSUB usar uma manipulação gerenciada, a prioridade para a mensagem será obtida do atributo **DefPriority** da fila modelo associada ao tópico inscrito.

### **MQPRI\_PRIORITY\_AS\_PUBLISHED**

A prioridade para a mensagem é a prioridade da publicação original. Este é o valor inicial do campo

Se alterar uma assinatura existente usando a opção MQSO\_ALTER, o *Priority* de quaisquer mensagens de publicação futuras poderá ser alterado.

No retorno de uma chamada MQSUB usando MQSO\_RESUME, esse campo é configurado para a prioridade atual sendo usada para a assinatura.

### **PubAccounting(MQBYTE32) para MQSD**

Esse é o valor que estará no campo *AccountingToken* do Message Descriptor (MQMD) de todas as mensagens de publicação correspondentes a essa assinatura. *AccountingToken* faz parte do contexto de identidade da mensagem.. Para obter mais informações sobre contexto da mensagem, consulte [Contexto da mensagem](#). Para obter mais informações sobre o campo *AccountingToken* no MQMD, consulte [“AccountingToken \(MQBYTE32\) para MQMD”](#) na página 475

É possível usar o seguinte valor especial para o campo *PubAccountingToken* :

#### **MQACT\_NONE**

Nenhum token de contabilidade especificado.

O valor é binário zero para o comprimento do campo.

Para a linguagem de programação C, a constante MQACT\_NONE\_ARRAY também é definida; ela tem o mesmo valor que MQACT\_NONE, mas é uma matriz de caracteres em vez de uma sequência.

Se a opção MQSO\_SET\_IDENTITY\_CONTEXT não for especificada, o token de contabilidade será gerado pelo gerenciador de filas como informações de contexto padrão e esse campo será um campo de saída que contém o *AccountingToken* que será configurado em cada mensagem publicada para essa assinatura

Se a opção MQSO\_SET\_IDENTITY\_CONTEXT for especificada, o token de contabilidade estará sendo gerado pelo usuário e esse campo será um campo de entrada que contém o *AccountingToken* a ser configurado em cada publicação para essa assinatura

O comprimento desse campo é fornecido por MQ\_ACCOUNTING\_TOKEN\_LENGTH. O valor inicial desse campo é MQACT\_NONE.

Se alterar uma assinatura existente usando a opção MQSO\_ALTER, o valor de *AccountingToken* em quaisquer mensagens de publicação futuras poderá ser alterado.

No retorno de uma chamada MQSUB usando MQSO\_RESUME, esse campo é configurado para o atual *AccountingToken* sendo usado para a assinatura.

### **PubApplIdentityData (MQCHAR32) para MQSD.**

Esse é o valor que está no campo *ApplIdentityData* do Message Descriptor (MQMD) de todas as mensagens de publicação correspondentes a essa assinatura. *ApplIdentityData* faz parte do contexto

de identidade da mensagem.. Para obter mais informações sobre contexto da mensagem, consulte [Contexto da mensagem](#). Para obter mais informações sobre o campo *ApplIdentityData* no MQMD, consulte [“Dados de ApplIdentity\(MQCHAR32\) para MQMD”](#) na página 477

Se a opção MQSO\_SET\_IDENTITY\_CONTEXT não for especificada, o *ApplIdentityData* que é configurado em cada mensagem publicada para essa assinatura será espaços em branco, como informações de contexto padrão

Se a opção MQSO\_SET\_IDENTITY\_CONTEXT for especificada, o *PubApplIdentityData* estará sendo gerado pelo usuário e esse campo será um campo de entrada que contém o *ApplIdentityData* a ser configurado em cada publicação para essa assinatura

O comprimento desse campo é fornecido por MQ\_APPL\_IDENTITY\_DATA\_LENGTH O valor inicial desse campo é a sequência nula em C e 32 caracteres em branco em outras linguagens de programação.

Se alterar uma assinatura existente usando a opção MQSO\_ALTER, o *ApplIdentityData* de quaisquer mensagens de publicação futuras poderá ser alterado.

No retorno de uma chamada MQSUB usando MQSO\_RESUME, esse campo é configurado para o atual *ApplIdentityData* sendo usado para a assinatura.

### ***SelectionString (MQCHARV) para MQSD***

Essa é a sequência usada para fornecer os critérios de seleção usados ao assinar mensagens de um tópico.

Esse campo de comprimento variável será retornado na saída de uma chamada MQSUB usando a opção MQSO\_RESUME, se um buffer for fornecido e também houver um comprimento de buffer positivo em VSBufSize. Se nenhum buffer for fornecido na chamada, apenas o comprimento da sequência de seleções será retornado no campo VSLength do MQCHARV Se o buffer fornecido for menor que o espaço necessário para retornar o campo, somente VSBufSize bytes serão retornados no buffer fornecido.

Se *SelectionString* for especificado incorretamente, de acordo com a descrição de como usar a estrutura [“MQCHARV-Sequência de Comprimento Variável”](#) na página 301 ou se exceder o comprimento máximo, a chamada falhará com o código de razão MQRC\_SELECTION\_STRING\_ERROR.

O uso de *SelectionString* é descrito em [Seletores](#)

### ***SubLevel (MQLONG) para MQSD***

Esse é o nível associado à assinatura. As publicações serão entregues a essa assinatura apenas se ela estiver no conjunto de assinaturas com o valor SubLevel mais alto menor ou igual ao PubLevel usado no momento da publicação. No entanto, se uma publicação tiver sido retida, ela não estará mais disponível para assinantes em níveis mais altos porque ela será publicada novamente em PubLevel 1.

O valor deve estar no intervalo de zero a 9 Zero é o nível mais baixo.

O valor inicial desse campo é 1.

Para obter mais informações, consulte [Interceptando publicações](#).

Se alterar uma assinatura existente usando a opção MQSO\_ALTER, o SubLevel não poderá ser alterado.

Combinar um SubLevel com um valor maior que 1 com a opção MQSO\_PUBLICATIONS\_ON\_REQUEST não é permitido.

No retorno de uma chamada MQSUB usando MQSO\_RESUME, esse campo é configurado para o nível atual sendo usado para a assinatura.

### ***Sequência ResObject(MQCHARV) para MQSD***

Este é o nome do objeto longo após o gerenciador de filas resolver o nome fornecido em *ObjectName*

Se o nome do objeto longo for fornecido em *ObjectString* e nada for fornecido em *ObjectName*, o valor retornado nesse campo será igual ao fornecido em *ObjectString*.

Se esse campo for omitido (que é `ResObjectString.VSBufSize` for zero), o `ResObjectString` não será retornado, mas o comprimento será retornado em `ResObjectString.VSLength`. Se o comprimento for menor que a Sequência `ResObjectIntegral`, ele será truncado e retornará quantos caracteres mais à direita puder caber no comprimento fornecido.

Se `ResObjectString` for especificado incorretamente, de acordo com a descrição de como usar a estrutura `MQCHARV` ou se exceder o comprimento máximo, a chamada falhará com o código de razão `MQRC_RES_OBJECT_STRING_ERROR`.

## MQSMPO-Configurar opções de propriedade de mensagem.

A estrutura **MQSMPO** permite que aplicativos especifiquem opções que controlam como as propriedades de mensagens são configuradas. A estrutura é um parâmetro de entrada na chamada de **MQSETMP**

### Disponibilidade

All IBM MQ systems and IBM MQ clients.

### Conjunto de caracteres e codificação

Os dados em **MQSMPO** devem estar no conjunto de caracteres do aplicativo e codificação do aplicativo (**MQENC\_NATIVE**).

### Campos

**Nota:** Na tabela a seguir, os campos são agrupados por uso e não alfabeticamente. Os tópicos filhos seguem a mesma sequência.

Tabela 531. Campos em MQSMPO		
Nome e descrição do campo	Nome da constante	Valor inicial (se houver) da constante
<u>StrucId</u> (identificador de estrutura)	MQSMPO_STRUC_ID	'SMPO'
<u>Versão</u> (número da versão da estrutura).	MQSMPO_VERSION_1	1
<u>Opções</u> (opções).	MQSMPO_NONE	0
<u>ValueEncoding</u> (codificação de valor de propriedade).	MQENC_NATIVE	Depende do ambiente
<u>ValueCCSID</u> (conjunto de caractere de valor da propriedade)..	MQCCSI_APPL	-3
<p><b>Notas:</b></p> <ol style="list-style-type: none"> <li>1. O valor Sequência nula ou espaços em branco indica a sequência nula em C e caracteres em branco em outras linguagens de programação.</li> <li>2. Na linguagem de programação C, a variável macro <code>MQSMPO_DEFAULT</code> contém os valores que são listados na tabela. Ele pode ser usado da seguinte maneira para fornecer valores iniciais para os campos na estrutura:</li> </ol> <pre>MQSMPO MySMPO = {MQSMPO_DEFAULT};</pre>		

### Declarações de idiomas

Declaração C para MQSMPO

```
typedef struct tagMQSMPO MQSMPO;
```

```

struct tagMQSMPO {
    MQCHAR4  StrucId;          /* Structure identifier */
    MQLONG   Version;         /* Structure version number */
    MQLONG   Options;        /* Options that control the action of MQSETMP */
    MQLONG   ValueEncoding;   /* Encoding of Value */
    MQLONG   ValueCCSID;     /* Character set identifier of Value */
};

```

### Declaração COBOL para MQSMPO

```

** MQSMPO structure
10 MQSMPO.
** Structure identifier
15 MQSMPO-STRUCID PIC X(4).
** Structure version number
15 MQSMPO-VERSION PIC S9(9) BINARY.
** Options that control the action of MQSETMP
15 MQSMPO-OPTIONS PIC S9(9) BINARY.
** Encoding of VALUE
15 MQSMPO-VALUEENCODING PIC S9(9) BINARY.
** Character set identifier of VALUE
15 MQSMPO-VALUECCSID PIC S9(9) BINARY.

```

### Declaração PL/I para MQSMPO

```

dcl
1 MQSMPO based,
3 StrucId      char(4),          /* Structure identifier */
3 Version      fixed bin(31),   /* Structure version number */
3 Options      fixed bin(31),   /* Options that control the action of MQSETMP */
3 ValueEncoding fixed bin(31), /* Encoding of Value */
3 ValueCCSID   fixed bin(31), /* Character set identifier of Value */

```

### Declaração do High Level Assembler para MQSMPO

```

MQSMPO          DSECT
MQSMPO_STRUCID  DS CL4 Structure identifier
MQSMPO_VERSION  DS F   Structure version number
MQSMPO_OPTIONS  DS F   Options that control the action of
*               MQSETMP
MQSMPO_VALUEENCODING DS F   Encoding of VALUE
MQSMPO_VALUECCSID DS F   Character set identifier of VALUE
MQSMPO_LENGTH   EQU *-MQSMPO
MQSMPO_AREA     DS CL(MQSMPO_LENGTH)

```

### **StrucId (MQCHAR4) para MQSMPO**

Este é o identificador de estrutura da estrutura de opções da propriedade de mensagens configurada. Ele é sempre um campo de entrada Seu valor é MQSMPO\_STRUC\_ID.

O valor deve ser:.

#### **MQSMPO\_STRUC\_ID**

Identificador para a estrutura de opções de propriedade de mensagem configurada.

Para a linguagem de programação C, a constante **MQSMPO\_STRUC\_ID\_ARRAY** também é definida.. Isso tem o mesmo valor que **MQSMPO\_STRUC\_ID**, mas é uma matriz de caracteres em vez de uma sequência..

### **Versão (MQLONG) para MQSMPO**

Este é o número da versão da estrutura; o valor deve ser:

#### **MQSMPO\_VERSION\_1**

Version-1 configure a estrutura de opções da propriedade de mensagens

A constante a seguir especifica o número da versão atual:

## **MQSMPO\_CURRENT\_VERSION**

Versão atual da estrutura de opções de propriedade de mensagem de conjunto.

Isto é sempre um campo de entrada. O valor inicial desse campo é **MQSMPO\_VERSION\_1**.

## **Opções (MQLONG) para MQSMPO**

### **Opções de localização**

As opções a seguir estão relacionadas ao local relativo da propriedade em comparação com o cursor da propriedade:

#### **MQSMPO\_SET\_FIRST**

Configura o valor da primeira propriedade que corresponde ao nome especificado ou, se não existir, inclui uma nova propriedade após todas as outras propriedades com uma hierarquia correspondente.

#### **MQSMPO\_SET\_PROP\_UNDER\_CURSOR**

Configura o valor da propriedade apontada pelo cursor de propriedade. A propriedade apontada pelo cursor da propriedade é aquela que foi consultada pela última vez usando a opção MQIMPO\_INQ\_FIRST ou MQIMPO\_INQ\_NEXT.

O cursor de propriedade é reconfigurado quando a manipulação de mensagem é reutilizada em uma chamada MQGET, ou quando a manipulação de mensagem é especificada no campo *MsgHandle* da estrutura MQGMO ou MQPMO em uma chamada MQPUT.

Se essa opção for usada quando o cursor de propriedade ainda não tiver sido estabelecido, ou se o ponteiro de propriedade para o cursor de propriedade tiver sido excluído, a chamada falhará com o código de conclusão MQCC\_FAILED e o código de razão MQRC\_PROPERTY\_NOT\_AVAILABLE.

#### **MQSMPO\_SET\_PROP\_BEFORE\_CURSOR**

Configura uma nova propriedade antes da propriedade apontada pelo cursor da propriedade. A propriedade apontada pelo cursor da propriedade é aquela que foi consultada pela última vez usando a opção MQIMPO\_INQ\_FIRST ou MQIMPO\_INQ\_NEXT.

O cursor de propriedade é reconfigurado quando a manipulação de mensagem é reutilizada em uma chamada MQGET, ou quando a manipulação de mensagem é especificada no campo *MsgHandle* da estrutura MQGMO ou MQPMO em uma chamada MQPUT.

Se essa opção for usada quando o cursor de propriedade ainda não tiver sido estabelecido, ou se o ponteiro de propriedade para o cursor de propriedade tiver sido excluído, a chamada falhará com o código de conclusão MQCC\_FAILED e o código de razão MQRC\_PROPERTY\_NOT\_AVAILABLE.

#### **MQSMPO\_SET\_PROP\_AFTER\_CURSOR**

Configura uma nova propriedade após a propriedade apontada pelo cursor da propriedade. A propriedade apontada pelo cursor da propriedade é aquela que foi consultada pela última vez usando a opção MQIMPO\_INQ\_FIRST ou MQIMPO\_INQ\_NEXT.

O cursor de propriedade é reconfigurado quando a manipulação de mensagem é reutilizada em uma chamada MQGET, ou quando a manipulação de mensagem é especificada no campo *MsgHandle* da estrutura MQGMO ou MQPMO em uma chamada MQPUT.

Se essa opção for usada quando o cursor de propriedade ainda não tiver sido estabelecido, ou se o ponteiro de propriedade para o cursor de propriedade tiver sido excluído, a chamada falhará com o código de conclusão MQCC\_FAILED e o código de razão MQRC\_PROPERTY\_NOT\_AVAILABLE.

#### **MQSMPO\_APPEND\_PROPERTY**

Faz com que uma nova propriedade seja incluída após todas as outras propriedades com uma hierarquia correspondente. Se existir pelo menos uma propriedade que corresponda ao nome especificado, uma nova propriedade será incluída no final após o final dessa lista de propriedades.

Essa opção permite a criação de uma lista de propriedades com o mesmo nome.

Se você não precisar de nenhuma das opções descritas, use a seguinte opção:

## **MQSMPO\_NONE**

Nenhuma opção especificada.

Isto é sempre um campo de entrada. O valor inicial desse campo é MQSMPO\_SET\_FIRST.

## **ValueEncoding (MQLONG) para MQSMPO**

A codificação do valor da propriedade a ser configurado se o valor for numérico.

Isto é sempre um campo de entrada. O valor inicial desse campo é MQENC\_NATIVE.

## **ValueCCSID (MQLONG) para MQSMPO**

O conjunto de caracteres do valor da propriedade a ser configurado se o valor for uma cadeia de caracteres.

Isto é sempre um campo de entrada. O valor inicial desse campo é MQCCSI\_APPL.

## **MQSRO-Opções de solicitação de assinatura**

A estrutura MQSRO permite que o aplicativo especifique opções que controlam como uma solicitação de subscrição é feita. A estrutura é um parâmetro de entrada / saída na chamada MQSUBRQ.

## **Disponibilidade**

A estrutura MQSRO está disponível nas plataformas a seguir:

-  AIX
-  IBM i
-  Linux
-  Windows
-  z/OS

e para o IBM MQ MQI clients conectado a esses sistemas

## **Versão**

A versão atual do MQSRO é MQSRO\_VERSION\_1.

## **Conjunto de caracteres e codificação**

Os dados no MQSRO devem estar no conjunto de caracteres fornecido pelo atributo do gerenciador de filas do **CodedCharSetId** e pela codificação do gerenciador de filas locais fornecido pelo MQENC\_NATIVE. No entanto, se o aplicativo estiver em execução como um cliente MQI MQ , a estrutura deverá estar no conjunto de caracteres e na codificação do cliente.

## **Campos**

**Nota:** Na tabela a seguir, os campos são agrupados por uso e não alfabeticamente. Os tópicos filhos seguem a mesma sequência.

<b>Nome e descrição do campo</b>	<b>Nome da constante</b>	<b>Valor inicial (se houver) da constante</b>
StrucId (identificador de estrutura)	MQSRO_STRUC_ID	'SRO~'

Nome e descrição do campo	Nome da constante	Valor inicial (se houver) da constante
Versão (número da versão da estrutura).	MQSRO_VERSION_1	1
Opções (opções).	MQSRO_NONE	0
NumPubs (número de publicações)	Nenhum	0
<p><b>Notas:</b></p> <ol style="list-style-type: none"> <li>1. O símbolo ~ representa um único caractere em branco.</li> <li>2. Na linguagem de programação C, a variável macroMQSRO_DEFAULT contém os valores listados na tabela. Ele pode ser usado da seguinte maneira para fornecer valores iniciais para os campos na estrutura:</li> </ol> <pre>MQSRO MySRO = {MQSRO_DEFAULT};</pre>		

## Declarações de idiomas

### Declaração C para MQSRO

```
typedef struct tagMQSRO MQSRO;
struct tagMQSRO {
    MQCHAR4   StrucId;           /* Structure identifier */
    MQLONG    Version;          /* Structure version number */
    MQLONG    Options;          /* Options that control the action of MQSUBRQ */
    MQLONG    NumPubs;          /* Number of publications sent */
    /* Ver:1 */
};
```

### Declaração COBOL para MQSRO

```
** MQSRO structure
10 MQSRO.
** Structure identifier
15 MQSRO-STRUCID          PIC X(4).
** Structure version number
15 MQSRO-VERSION         PIC S9(9) BINARY.
** Options that control the action of MQSUBRQ
15 MQSRO-OPTIONS         PIC S9(9) BINARY.
** Number of publications sent
15 MQSRO-NUMPUBS         PIC S9(9) BINARY.
```

### Declaração PL/I para MQSRO

```
dcl
1 MQSRO based,
3 StrucId          char(4),          /* Structure identifier */
3 Version          fixed bin(31),    /* Structure version number */
3 Options          fixed bin(31),    /* Options that control the action of MQSUBRQ */
3 NumPubs          fixed bin(31);    /* Number of publications sent */
```

### Declaração do High Level Assembler para MQSRO

```
MQSRO
MQSRO_STRUCID      DSECT
MQSRO_VERSION      DS    CL4   Structure identifier
MQSRO_OPTIONS      DS    F     Structure version number
MQSRO_NUMPUBS      DS    F     Options that control the action of MQSUBRQ
*
MQSRO_LENGTH       EQU    *-MQSRO
MQSRO_AREA         ORG    MQSRO
                   DS     CL(MQSRO_LENGTH)
```



### ***StrucId (MQCHAR4) para MQSRO***

Este é o identificador de estrutura da estrutura de opções de solicitação de assinatura. Ele é sempre um campo de entrada Seu valor é MQSRO\_STRUC\_ID.

O valor deve ser:.

#### **MQSRO\_STRUC\_ID**

Identificador para a estrutura de opções de solicitação de assinatura.

Para a linguagem de programação C, a constante MQSRO\_STRUC\_ID\_ARRAY também é definida Ele tem o mesmo valor que MQSRO\_STRUC\_ID, mas é uma matriz de caracteres em vez de uma sequência.

### ***Versão (MQLONG) para MQSRO***

Este é o número da versão da estrutura; o valor deve ser:

#### **MQSRO\_VERSION\_1**

Version-1 Estrutura de Opções de Solicitação de Assinatura.

A constante a seguir especifica o número da versão atual:

#### **MQSRO\_CURRENT\_VERSION**

Versão atual da estrutura de Opções de Solicitação de Assinatura

Isto é sempre um campo de entrada. O valor inicial desse campo é MQSRO\_VERSION\_1.

### ***Opções (MQLONG) para MQSRO***

Uma das opções a seguir deve ser especificada: Apenas uma opção pode ser especificada

#### **MQSRO\_FAIL\_IF QUIESCING**

A chamada MQSUBRQ falhará se o gerenciador de fila estiver no estado de quiesce No z/OS, para um aplicativo CICS ou IMS , essa opção também força a chamada MQSUBRQ a falhar se a conexão estiver em um estado de quiesce

**Opção padrão:** Se a opção descrita anteriormente não for necessária a opção a seguir deve ser usada:

#### **MQSRO\_NONE**

Use esse valor para indicar que nenhuma outra opção foi especificada. Todas as opções assumem seus valores-padrão.

MQSRO\_NONE ajuda a documentação do programa.. Embora não seja pretendido que essa opção seja usada com qualquer outra, porque seu valor é zero, esse uso não pode ser detectado..

### ***NumPubs (MQLONG) para MQSRO***

Este é um campo de saída, retornado para o aplicativo para indicar o número de publicações enviadas para a fila de assinaturas como resultado dessa chamada Embora esse número de publicações tenha sido enviado como resultado dessa chamada, não há garantia de que essas muitas mensagens estarão disponíveis para o aplicativo obter, especialmente se forem mensagens não persistentes.

Pode haver mais de uma publicação se o tópico inscrito continha um curinga. Se nenhum curinga estava presente na cadeia de tópicos quando a assinatura representada por *Hsub* foi criada, no máximo uma publicação será enviada como resultado dessa chamada.

### **MQSTS-Estrutura de relatório de status.**

A estrutura MQSTS é um parâmetro de saída do comando MQSTAT. O comando MQSTAT é usado para recuperar informações de status.. Essas informações são retornadas na estrutura MQSTS.

## Conjunto de caracteres e codificação

Os dados de caracteres em MQSTS estão no conjunto de caracteres do gerenciador de fila local; isso é fornecido pelo atributo do gerenciador de filas *CodedCharSetId* .. Os dados numéricos no MQSTS estão na codificação da máquina nativa; isso é fornecido pela *Codificação*

## Campos

**Nota:** Na tabela a seguir, os campos são agrupados por uso e não alfabeticamente. Os tópicos filhos seguem a mesma sequência.

<i>Tabela 532. Campos em MQSTS</i>		
Nome e descrição do campo	Nome da constante	Valor inicial (se houver) da constante
<u>StrucId</u> (identificador de estrutura)	MQSTS_STRUC_ID	'STAT-'
<u>Versão</u> (número da versão da estrutura).	MQSTS_VERSION_1	1
<u>CompCode</u> (código de conclusão do primeiro erro).	MQCC_OK	0
<u>Razão</u> (código de razão do primeiro erro)..	MQRC_NONE	0
<u>PutSuccessContagem</u> (número de chamadas put assíncronas bem-sucedidas)	Nenhum	0
<u>PutWarningCount</u> (número de chamadas put assíncronas que tinham avisos)	Nenhum	0
<u>PutFailureContagem</u> (número de chamadas put assíncronas com falha)	Nenhum	0
<u>ObjectType</u> (tipo de objeto com falha).	MQOT_Q	1
<u>ObjectName</u> (nome do objeto com falha).	Nenhum	Cadeia nula ou espaços em branco
<u>ObjectQMgrNome</u> (nome do gerenciador de filas que possui o objeto com falha)	Nenhum	Cadeia nula ou espaços em branco
<u>ResolvedObjectName</u> (nome resolvido da fila de destino);	Nenhum	Cadeia nula ou espaços em branco
<u>ResolvedQMgrNome</u> (nome resolvido do gerenciador de filas de destino)	Nenhum	Cadeia nula ou espaços em branco
<b>Nota:</b> Os campos restantes serão ignorados se a Versão for menor que MQSTS_VERSION_2.		
<u>ObjectString</u> (nome longo do objeto com falha)	MQCHARV_DEFAULT	{NULL,0,0,0,-3}
<u>SubName</u> (nome da assinatura com falha)	MQCHARV_DEFAULT	{NULL,0,0,0,-3}
<u>OpenOptions</u> (opções abertas associadas à falha)	Nenhum	0
<u>SubOptions</u> (opções de assinatura associadas à falha)	Nenhum	0

Tabela 532. Campos em MQSTS (continuação)

Nome e descrição do campo	Nome da constante	Valor inicial (se houver) da constante
<b>Notas:</b>		
<ol style="list-style-type: none"> <li>1. O símbolo <code>\n</code> representa um único caractere em branco.</li> <li>2. O valor Sequência nula ou espaços em branco indica a sequência nula em C e caracteres em branco em outras linguagens de programação.</li> <li>3. Na linguagem de programação C, a variável de macro <code>MQSTS_DEFAULT</code> contém os valores listados na tabela. Ele pode ser usado da seguinte maneira para fornecer valores iniciais para os campos na estrutura: <div style="background-color: #f0f0f0; padding: 5px; margin-top: 10px;"> <pre>MQSTS MySTS = {MQSTS_DEFAULT};</pre> </div> </li> </ol>		

## Declarações de idiomas

### Declaração C para MQSTS

```
typedef struct tagMQSTS MQSTS;
struct tagMQSTS {
    MQCHAR4   StructId;           /* Structure identifier */
    MQLONG    Version;           /* Structure version number */
    MQLONG    CompCode;          /* Completion Code of first error */
    MQLONG    Reason;            /* Reason Code of first error */
    MQLONG    PutSuccessCount;    /* Number of Async calls succeeded */
    MQLONG    PutWarningCount;    /* Number of Async calls had warnings */
    MQLONG    PutFailureCount;    /* Number of Async calls had failures */
    MQLONG    ObjectType;        /* Failing object type */
    MQCHAR48  ObjectName;        /* Failing object name */
    MQCHAR48  ObjectQMgrName;     /* Failing object queue manager name */
    MQCHAR48  ResolvedObjectName; /* Resolved name of destination queue */
    MQCHAR48  ResolvedQMgrName;  /* Resolved name of destination qmgr */
    /* Ver:1 */
    MQCHARV   ObjectString;       /* Failing object long name */
    MQCHARV   SubName;            /* Failing subscription name */
    MQLONG    OpenOptions;        /* Failing open options */
    MQLONG    SubOptions;         /* Failing subscription options */
    /* Ver:2 */
};
```

### Declaração COBOL para MQSTS

```
** MQSTS structure
  10 MQSTS.
** Structure identifier
  15 MQSTS-STRUCID PIC X(4).
** Structure version number
  15 MQSTS-VERSION PIC S9(9) BINARY.
** Completion Code of first error
  15 MQSTS-COMPCODE PIC S9(9) BINARY.
** Reason Code of first error
  15 MQSTS-REASON PIC S9(9) BINARY.
** Number of Async put calls succeeded
  15 MQSTS-PUTSUCCESSCOUNT PIC S9(9) BINARY.
** Number of Async put calls had warnings
  15 MQSTS-PUTWARNINGCOUNT PIC S9(9) BINARY.
** Number of Async put calls had failures
  15 MQSTS-PUTFAILURECOUNT PIC S9(9) BINARY.
** Failing object type
  15 MQSTS-OBJECTTYPE PIC S9(9) BINARY.
** Failing object name
  15 MQSTS-OBJECTNAME PIC X(48).
** Failing object queue manager
  15 MQSTS-OBJECTQMGRNAME PIC X(48).
** Resolved name of destination queue
  15 MQSTS-RESOLVEDOBJECTNAME PIC X(48).
```

```

** Resolved name of destination qmgr
 15 MQSTS-RESOLVEDQMGRNAME PIC X(48).
** Ver:1 **
** Failing object long name
 15 MQSTS-OBJECTSTRING.
** Address of variable length string
 20 MQSTS-OBJECTSTRING-VSPTR POINTER.
** Offset of variable length string
 20 MQSTS-OBJECTSTRING-VSOFFSET PIC S9(9) BINARY.
** Size of buffer
 20 MQSTS-OBJECTSTRING-VSBUFSIZE PIC S9(9) BINARY.
** Length of variable length string
 20 MQSTS-OBJECTSTRING-VSLENGTH PIC S9(9) BINARY.
** CCSID of variable length string
 20 MQSTS-OBJECTSTRING-VSCCSID PIC S9(9) BINARY.
** Failing subscription name
 15 MQSTS-SUBNAME.
** Address of variable length string
 20 MQSTS-SUBNAME-VSPTR POINTER.
** Offset of variable length string
 20 MQSTS-SUBNAME-VSOFFSET PIC S9(9) BINARY.
** Size of buffer
 20 MQSTS-SUBNAME-VSBUFSIZE PIC S9(9) BINARY.
** Length of variable length string
 20 MQSTS-SUBNAME-VSLENGTH PIC S9(9) BINARY.
** CCSID of variable length string
 20 MQSTS-SUBNAME-VSCCSID PIC S9(9) BINARY.
** Failing open options
 15 MQSTS-OPENOPTIONS PIC S9(9) BINARY.
** Failing subscription options
 15 MQSTS-SUBOPTIONS PIC S9(9) BINARY.
** Ver:2 **

```

## Declaração PL/I para MQSTS

```

dcl
 1 MQSTS based,
 3 StrucId          char(4),          /* Structure identifier */
 3 Version         fixed bin(31),    /* Structure version number */
 3 CompCode       fixed bin(31),    /* Completion code */
 3 Reason         fixed bin(31),    /* Reason code */
 3 PutSuccessCount fixed bin(31),    /* Put success count */
 3 PutWarningCount fixed bin(31),    /* Put warning count */
 3 PutFailureCount fixed bin(31),    /* Put failure count */
 3 ObjectType     fixed bin(31),    /* Object type */
 3 ObjectName     char(48),          /* Object name */
 3 ObjectQmgrName char(48),          /* Object queue manager */
 3 ResolvedObjectName char(48),    /* Resolved Object name */
 3 ResolvedQmgrName char(48);      /* Resolved Object queue manager */
/* Ver:1 */
 3 ObjectString,          /* Failing object long name */
 5 VSPtr pointer,        /* Address of variable length string */
 5 VSOffset fixed bin(31), /* Offset of variable length string */
 5 VSBufSize fixed bin(31), /* Size of buffer */
 5 VSLength fixed bin(31), /* Length of variable length string */
 5 VSCCSID fixed bin(31); /* CCSID of variable length string */
 3 SubName,              /* Failing subscription name */
 5 VSPtr pointer,        /* Address of variable length string */
 5 VSOffset fixed bin(31), /* Offset of variable length string */
 5 VSBufSize fixed bin(31), /* Size of buffer */
 5 VSLength fixed bin(31), /* Length of variable length string */
 5 VSCCSID fixed bin(31); /* CCSID of variable length string */
 3 OpenOptions fixed bin(31), /* Failing open options */
 3 SubOptions fixed bin(31); /* Failing subscription options */
/* Ver:2 */

```

## Declaração do High Level Assembler para MQSTS

MQSTS	DSECT		
MQSTS_STRUCID	DS	CL4	Structure identifier
MQSTS_VERSION	DS	F	Structure version number
MQSTS_COMPCODE	DS	F	Completion code
MQSTS_REASON	DS	F	Reason code
MQSTS_PUTSUCCESSCOUNT	DS	F	Success count
MQSTS_PUTWARNINGCOUNT	DS	F	Warning count
MQSTS_PUTFAILURECOUNT	DS	F	Failure count
MQSTS_OBJTYPE	DS	F	Object type

MQSTS_OBJNAME	DS	CL48	Object name
MQSTS_OBJQMGR	DS	CL48	Object queue manager
MQSTS_ROBJNAME	DS	CL48	Resolved object name
MQSTS_ROBJQMGR	DS	CL48	Resolved object queue manager
MQSTS_OBJECTSTRING	DS	0F	Force fullword alignment
MQSTS_OBJECTSTRING_VSPTR	DS	A	Address of variable length string
MQSTS_OBJECTSTRING_VSOFFSET	DS	F	Offset of variable length string
MQSTS_OBJECTSTRING_VSBUFSIZE	DS	F	Size of buffer
MQSTS_OBJECTSTRING_VSLENGTH	DS	F	Length of variable length string
MQSTS_OBJECTSTRING_VSCCSID	DS	F	CCSID of variable length string
MQSTS_OBJECTSTRING_LENGTH	EQU	*	MQSTS_OBJECTSTRING
		ORG	MQSTS_OBJECTSTRING
MQSTS_OBJECTSTRING_AREA	DS	CL	(MQSTS_OBJECTSTRING_LENGTH)
*			
MQSTS_SUBNAME	DS	0F	Force fullword alignment
MQSTS_SUBNAME_VSPTR	DS	A	Address of variable length string
MQSTS_SUBNAME_VSOFFSET	DS	F	Offset of variable length string
MQSTS_SUBNAME_VSBUFSIZE	DS	F	Size of buffer
MQSTS_SUBNAME_VSLENGTH	DS	F	Length of variable length string
MQSTS_SUBNAME_VSCCSID	DS	F	CCSID of variable length string
MQSTS_SUBNAME_LENGTH	EQ	*	MQSTS_SUBNAME
		ORG	MQSTS_SUBNAME
MQSTS_SUBNAME_AREA	DS	CL	(MQSTS_SUBNAME_LENGTH)
*			
MQSTS_OPENOPTIONS	DS	F	Failing open options
MQSTS_SUBOPTIONS	DS	F	Failing subscription option
MQSTS_LENGTH	EQU	*	MQSTS
	ORG		MQSTS
MQSTS_AREA	DS	CL	(MQSTS_LENGTH)

### Referências relacionadas

“MQSTAT-Recuperar informações de status” na página 812

Use a chamada MQSTAT para recuperar as informações de status. O tipo de informações de status retornadas é determinado pelo valor de Tipo especificado na chamada.

### StrucId (MQCHAR4) para MQSTS

Este é o identificador da estrutura de relatório de status. Ele é sempre um campo de entrada. Seu valor é MQSTS\_STRUC\_ID.

O valor deve ser::

#### MQSTS\_STRUC\_ID

Identificador para a estrutura de relatório de status

Para a linguagem de programação C, a constante MQSTS\_STRUC\_ID\_ARRAY também é definida.. Isso tem o mesmo valor que MQSTS\_STRUC\_ID, mas é uma matriz de caracteres em vez de uma sequência..

### Versão (MQLONG) para MQSTS

O número da versão da estrutura

O valor deve ser:

#### MQSTS\_VERSION\_1

Estrutura de relatório de status da Versão 1

#### MQSTS\_VERSION\_2

Estrutura de relatório de status da versão 2

A constante a seguir especifica o número da versão atual:

#### MQSTS\_CURRENT\_VERSION

Versão atual da estrutura de relatório de status. A versão atual é MQSTS\_VERSION\_2..

Version é sempre um campo de entrada.. Seu valor inicial é MQSTS\_VERSION\_1..

### CompCode (MQLONG) para MQSTS

O código de conclusão da operação que está sendo relatado.

A interpretação de CompCode depende do valor do parâmetro MQSTAT **Type** .

### **MQSTAT\_TYPE\_ASYNC\_ERROR**

Este é o código de conclusão resultante de uma operação de colocação assíncrona anterior no objeto especificado em `ObjectName`

### **MQSTAT\_TYPE\_RECONNECTION**

Se a conexão estiver reconectando ou falhou ao reconectar, este será o código de conclusão que fez com que a conexão começasse a se reconectar

Se a conexão estiver conectada no momento, o valor será `MQCC_OK`

### **MQSTAT\_TYPE\_RECONNECTION\_ERROR**

Se a conexão falhou ao reconectar, este é o código de conclusão que causou a falha da reconexão.

Se a conexão estiver conectada atualmente, ou reconectando, o valor será `MQCC_OK`

`CompCode` é sempre um campo de saída.. Seu valor inicial é `MQCC_OK`..

### ***Razão (MQLONG) para MQSTS***

O código de razão da operação em que está sendo relatado

A interpretação de Reason depende do valor do parâmetro **MQSTAT Type** .

### **MQSTAT\_TYPE\_ASYNC\_ERROR**

Este é o código de razão resultante de uma operação de colocação assíncrona anterior no objeto especificado em `ObjectName`

### **MQSTAT\_TYPE\_RECONNECTION**

Se a conexão estiver reconectando ou falhou ao reconectar, este é o código de razão que fez com que a reconexão começasse a se reconectar

Se a conexão estiver conectada no momento, o valor será `MQRC_NONE`

### **MQSTAT\_TYPE\_RECONNECTION\_ERROR**

Se a conexão falhou ao reconectar, este é o código de razão que causou a falha da reconexão.

Se a conexão estiver conectada atualmente, ou reconectando, o valor será `MQRC_NONE`

Reason é um campo de saída.. Seu valor inicial é `MQRC_NONE`..

### ***Contagem de PutSuccess(MQLONG) para MQSTS***

O número de operações de colocação assíncronas bem-sucedidas.

O valor de `PutSuccessCount` depende do valor do parâmetro **MQSTAT Type** .

### **MQSTAT\_TYPE\_ASYNC\_ERROR**

O número de operações put assíncronas no objeto nomeado na estrutura `MQSTS` que foi concluído com `MQCC_OK`.

### **MQSTAT\_TYPE\_RECONNECTION**

Zero.

### **MQSTAT\_TYPE\_RECONNECTION\_ERROR**

Zero.

`PutSuccessCount` é um campo de saída.. Seu valor inicial é zero.

### ***Contagem de PutWarning(MQLONG) para MQSTS***

O número de operações put assíncronas que terminaram com um aviso.

O valor de `PutWarningCount` depende do valor do parâmetro **MQSTAT Type** .

### **MQSTAT\_TYPE\_ASYNC\_ERROR**

O número de operações put assíncronas no objeto nomeado na estrutura MQSTS que foi concluído com MQCC\_WARNING.

### **MQSTAT\_TYPE\_RECONNECTION**

Zero.

### **MQSTAT\_TYPE\_RECONNECTION\_ERROR**

Zero.

PutWarningCount é um campo de saída.. Seu valor inicial é zero.

### **PutFailureContagem (MQLONG) para MQSTS**

O número de operações de colocação assíncronas que falharam

O valor de PutFailureCount depende do valor do parâmetro MQSTAT **Type** .

### **MQSTAT\_TYPE\_ASYNC\_ERROR**

O número de operações put assíncronas no objeto nomeado na estrutura MQSTS que foi concluído com MQCC\_FAILED.

### **MQSTAT\_TYPE\_RECONNECTION**

Zero.

### **MQSTAT\_TYPE\_RECONNECTION\_ERROR**

Zero.

PutFailureCount é um campo de saída.. Seu valor inicial é zero.

### **ObjectType (MQLONG) para MQSTS**

O tipo do objeto nomeado em *ObjectName* sendo relatado.

Os valores possíveis de ObjectType são listados em [“MQOT\\_\\* \(Tipos de Objeto e Tipos de Objeto Estendido\)” na página 165](#)

ObjectType é um campo de saída.. Seu valor inicial é MQOT\_Q..

### **ObjectName (MQCHAR48) para MQSTS**

O nome do objeto que está sendo relatado.

A interpretação de ObjectName depende do valor do parâmetro MQSTAT **Type** .

### **MQSTAT\_TYPE\_ASYNC\_ERROR**

Este é o nome da fila ou tópico usado na operação put, cuja falha é relatada nos campos *CompCode* e *Reason* na estrutura MQSTS .

### **MQSTAT\_TYPE\_RECONNECTION**

Se a conexão estiver se reconectando, esse será o nome do gerenciador de filas associado à conexão

### **MQSTAT\_TYPE\_RECONNECTION\_ERROR**

Se a conexão falhou ao se reconectar, este é o nome do objeto que causou a falha da reconexão A razão para a falha é relatada nos campos *CompCode* e *Reason* na estrutura MQSTS

ObjectName é um campo de saída.. Seu valor inicial é a sequência nula em C e 48 caracteres em branco em outras linguagens de programação.

### **ObjectQMgrNome (MQCHAR48) para MQSTS**

O nome do gerenciador de filas sendo relatado.

A interpretação de ObjectQMgrName depende do valor do parâmetro MQSTAT **Type** .

## **MQSTAT\_TYPE\_ASYNC\_ERROR**

Este é o nome do gerenciador de filas no qual o objeto *ObjectName* está definido Um nome que está inteiramente em branco até o primeiro caractere nulo ou o final do campo denota o gerenciador de fila ao qual o aplicativo está conectado (o gerenciador de filas locais)...

## **MQSTAT\_TYPE\_RECONNECTION**



O campo **ObjectQMgrName** contém o nome de um gerenciador de filas ao qual a reconexão está sendo solicitada ou em branco se nenhum gerenciador de filas for especificado. Se possível, o cliente tenta se reconectar a um gerenciador de filas com esse nome.



em branco.

## **MQSTAT\_TYPE\_RECONNECTION\_ERROR**

Se a conexão falhou ao se reconectar, este é o nome do objeto que causou a falha da reconexão A razão para a falha é relatada nos campos *CompCode* e *Reason* na estrutura MQSTS

*ObjectQMgrName* é um campo de saída.. Seu valor é a sequência nula em C e 48 caracteres em branco em outras linguagens de programação.

### ***ResolvedObjectName (MQCHAR48) para MQSTS***

O nome do objeto nomeado em *ObjectName* após o gerenciador de fila local resolver o nome..

A interpretação de *ResolvedObjectName* depende do valor do parâmetro **MQSTAT Type** .

## **MQSTAT\_TYPE\_ASYNC\_ERROR**

*ResolvedObjectName* é o nome do objeto nomeado em *ObjectName* após o gerenciador de fila local resolver o nome O nome retornado é o nome de um objeto que existe no gerenciador de filas identificado por *ResolvedQMgrName*.

## **MQSTAT\_TYPE\_RECONNECTION**

em branco.

## **MQSTAT\_TYPE\_RECONNECTION\_ERROR**

em branco.

*ResolvedObjectName* é um campo de saída.. Seu valor inicial é a sequência nula em C e 48 caracteres em branco em outras linguagens de programação.

### ***ResolvedQMgrName (MQCHAR48) para MQSTS***

O nome do gerenciador de filas de destino após o gerenciador de fila local resolver o nome..

A interpretação de *ResolvedQMgrName* depende do valor do parâmetro **MQSTAT Type** .

## **MQSTAT\_TYPE\_ASYNC\_ERROR**

*ResolvedQMgrName* é o nome do gerenciador de filas de destino após o gerenciador de fila local resolver o nome.. O nome retornado é o nome do gerenciador de filas que possui o objeto identificado por *ResolvedObjectName*.. *ResolvedQMgrName* pode ser o nome do gerenciador de filas locais..

## **MQSTAT\_TYPE\_RECONNECTION**

em branco.

## **MQSTAT\_TYPE\_RECONNECTION\_ERROR**

em branco.

*ResolvedQMgrName* é sempre um campo de saída.. Seu valor inicial é a sequência nula em C e 48 caracteres em branco em outras linguagens de programação.



### ***ObjectString (MQCHARV) para MQSTS***

Nome do objeto longo do objeto com falha que está sendo relatado. Presente apenas na versão 2 de MQSTS ou superior.

A interpretação de ObjectString depende do valor do parâmetro MQSTAT **Type** .

#### **MQSTAT\_TYPE\_ASYNC\_ERROR**

Este é o nome do objeto longo da fila ou do tópico usado na operação MQPUT , que falhou

#### **MQSTAT\_TYPE\_RECONNECTION**

Cadeia de comprimento zero

#### **MQSTAT\_TYPE\_RECONNECTION\_ERROR**

Este é o nome longo do objeto que causou a falha da reconexão.

ObjectString é um campo de saída.. Seu valor inicial é uma cadeia de comprimento zero.

### ***SubName (MQCHARV) para MQSTS***

O nome da assinatura com falha.. Presente apenas na versão 2 de MQSTS ou superior.

A interpretação de SubName depende do valor do parâmetro MQSTAT **Type** .

#### **MQSTAT\_TYPE\_ASYNC\_ERROR**

Sequência de comprimento zero.

#### **MQSTAT\_TYPE\_RECONNECTION**

Sequência de comprimento zero.

#### **MQSTAT\_TYPE\_RECONNECTION\_ERROR**

O nome da assinatura que causou a falha de reconexão. Se nenhum nome de assinatura estiver disponível ou a falha não estiver relacionada a uma assinatura, esta será uma sequência de comprimento zero.

SubName é um campo de saída.. Seu valor inicial é uma cadeia de comprimento zero.

### ***OpenOptions (MQLONG) para MQSTS***

O OpenOptions usado para abrir o objeto que está sendo relatado. Presente apenas na versão 2 de MQSTS ou superior.

O valor de OpenOptions depende do valor do parâmetro MQSTAT **Type** .

#### **MQSTAT\_TYPE\_ASYNC\_ERROR**

Zero.

#### **MQSTAT\_TYPE\_RECONNECTION**

Zero.

#### **MQSTAT\_TYPE\_RECONNECTION\_ERROR**

O OpenOptions usado quando a falha ocorreu.. A razão para a falha é relatada nos campos *CompCode* e *Reason* na estrutura MQSTS

OpenOptions é um campo de saída.. Seu valor inicial é zero.

### ***SubOptions (MQLONG) para MQSTS***

O SubOptions usado para abrir a assinatura com falha.. Presente apenas na versão 2 de MQSTS ou superior.

A interpretação de SubOptions depende do valor do parâmetro MQSTAT **Type** .

#### **MQSTAT\_TYPE\_ASYNC\_ERROR**

Zero.

## **MQSTAT\_TYPE\_RECONNECTION**

Zero.

## **MQSTAT\_TYPE\_RECONNECTION\_ERROR**

O SubOptions usado quando a falha ocorreu.. Se a falha não estiver relacionada à assinatura para um tópico, o valor retornado será zero.

SubOptions é um campo de saída.. Seu valor inicial é zero.

## **MQTM-Mensagem do acionador**

A estrutura MQTM descreve os dados na mensagem do acionador que é enviada pelo gerenciador de fila para um aplicativo de monitor acionador quando um evento acionador ocorre para uma fila. Essa estrutura faz parte da IBM MQ Trigger Monitor Interface (TMI), que é uma das interfaces de estrutura IBM MQ .

### **Nome do Formato**

MQFMT\_TRIGGER.

### **Conjunto de caracteres e codificação**



Os dados de caractere no MQTM estão no conjunto de caracteres do gerenciador de filas que gera o MQTM. Os dados numéricos no MQTM estão na codificação de máquina do gerenciador de filas que gera o MQTM.

O conjunto de caracteres e a codificação do MQTM são fornecidos pelos campos *CodedCharSetId* e *Encoding* em:

- O MQMD (se a estrutura MQTM estiver no início dos dados da mensagem) ou
- A estrutura do cabeçalho que precede a estrutura MQTM (todos os outros casos).

### **Uso**

Um aplicativo de monitor acionador pode precisar passar algumas ou todas as informações na mensagem do acionador para o aplicativo que o aplicativo de monitor acionador inicia.. As informações que podem ser necessárias para o aplicativo iniciado incluem *QName*, *TriggerData* e *UserData*. O aplicativo de monitor acionador pode transmitir a estrutura MQTM diretamente para o aplicativo iniciado ou transmitir uma estrutura MQTMC2 , dependendo do que é permitido pelo ambiente e conveniente para o aplicativo iniciado. Para obter informações sobre MQTMC2, consulte [“MQTMC2 -Mensagem do acionador 2 \(formato de caractere\)”](#) na página 629.

-  No z/OS, para um aplicativo MQAT\_CICS que é iniciado usando a transação CKTI, a estrutura de mensagem do acionador inteira do MQTM é disponibilizada para a transação iniciada; as informações podem ser recuperadas usando o comando EXEC CICS RETRIEVE
-  No IBM i, o aplicativo do monitor acionador fornecido com IBM MQ passa uma estrutura MQTMC2 para o aplicativo iniciado.

Para obter informações sobre como usar acionadores, consulte [Iniciando IBM MQ aplicativos que usam acionadores](#)

### **Campos**

**Nota:** Na tabela a seguir, os campos são agrupados por uso e não alfabeticamente. Os tópicos filhos seguem a mesma sequência.

Tabela 533. Campos em MQTM para MQTM

Nome e descrição do campo	Nome da constante	Valor inicial (se houver) da constante
StrucId (identificador de estrutura)	MQTM_STRUC_ID	'TM↵↵'
Versão (número da versão da estrutura).	MQTM_VERSION_1	1
QName (nome da fila acionada)	Nenhum	Cadeia nula ou espaços em branco
ProcessName (nome do objeto de processo)	Nenhum	Cadeia nula ou espaços em branco
TriggerData (dados do acionador)	Nenhum	Cadeia nula ou espaços em branco
ApplType (tipo de aplicativo)	Nenhum	0
ApplId (identificador do aplicativo).	Nenhum	Cadeia nula ou espaços em branco
EnvData (dados do ambiente)	Nenhum	Cadeia nula ou espaços em branco
UserData (dados do usuário)	Nenhum	Cadeia nula ou espaços em branco

**Notas:**

1. O símbolo ↵ representa um único caractere em branco.
2. O valor Sequência nula ou espaços em branco indica a sequência nula em C e caracteres em branco em outras linguagens de programação.
3. Na linguagem de programação C, a variável macroMQTM\_DEFAULT contém os valores listados na tabela. Use-o da seguinte maneira para fornecer valores iniciais para os campos na estrutura:

```
MQTM MyTM = {MQTM_DEFAULT};
```

**Declarações de idiomas**

Declaração C para MQTM

```
typedef struct tagMQTM MQTM;
struct tagMQTM {
    MQCHAR4    StrucId;        /* Structure identifier */
    MQLONG    Version;        /* Structure version number */
    MQCHAR48   QName;         /* Name of triggered queue */
    MQCHAR48   ProcessName;   /* Name of process object */
    MQCHAR64   TriggerData;   /* Trigger data */
    MQLONG    ApplType;       /* Application type */
    MQCHAR256  ApplId;        /* Application identifier */
    MQCHAR128  EnvData;       /* Environment data */
    MQCHAR128  UserData;      /* User data */
};
```

Declaração COBOL para MQTM

```
** MQTM structure
10 MQTM.
** Structure identifier
15 MQTM-STRUCID PIC X(4).
** Structure version number
15 MQTM-VERSION PIC S9(9) BINARY.
```

```

**   Name of triggered queue
15 MQTM-QNAME      PIC X(48).
**   Name of process object
15 MQTM-PROCESSNAME PIC X(48).
**   Trigger data
15 MQTM-TRIGGERDATA PIC X(64).
**   Application type
15 MQTM-APPLTYPE   PIC S9(9) BINARY.
**   Application identifier
15 MQTM-APPLID     PIC X(256).
**   Environment data
15 MQTM-ENVDATA    PIC X(128).
**   User data
15 MQTM-USERDATA   PIC X(128).

```

### Declaração PL/I para MQTM

```

dcl
  1 MQTM based,
    3 StrucId      char(4),          /* Structure identifier */
    3 Version     fixed bin(31),    /* Structure version number */
    3 QName       char(48),         /* Name of triggered queue */
    3 ProcessName char(48),         /* Name of process object */
    3 TriggerData char(64),         /* Trigger data */
    3 ApplType    fixed bin(31),    /* Application type */
    3 ApplId     char(256),         /* Application identifier */
    3 EnvData     char(128),        /* Environment data */
    3 UserData    char(128);        /* User data */

```

### Declaração do High Level Assembler para MQTM

```

MQTM          DSECT
MQTM_STRUCID  DS   CL4   Structure identifier
MQTM_VERSION  DS   F     Structure version number
MQTM_QNAME    DS   CL48  Name of triggered queue
MQTM_PROCESSNAME DS CL48  Name of process object
MQTM_TRIGGERDATA DS CL64  Trigger data
MQTM_APPLTYPE DS   F     Application type
MQTM_APPLID   DS   CL256 Application identifier
MQTM_ENVDATA  DS   CL128 Environment data
MQTM_USERDATA DS   CL128 User data
*
MQTM_LENGTH   EQU   *-MQTM
ORG   MQTM
MQTM_AREA     DS   CL(MQTM_LENGTH)

```

### Declaração do Visual Basic para MQTM

```

Type MQTM
  StrucId      As String*4   'Structure identifier'
  Version     As Long        'Structure version number'
  QName       As String*48   'Name of triggered queue'
  ProcessName As String*48   'Name of process object'
  TriggerData As String*64   'Trigger data'
  ApplType    As Long        'Application type'
  ApplId     As String*256   'Application identifier'
  EnvData     As String*128  'Environment data'
  UserData    As String*128  'User data'
End Type

```

## MQMD para uma mensagem do acionador

Tabela 534. Configurações para os campos no MQMD de uma mensagem acionador gerada pelo gerenciador de filas

Campo no MQMD	Valor Usado
<i>StrucId</i>	MQMD_STRUC_ID
<i>Version</i>	MQMD_VERSION_1

Tabela 534. Configurações para os campos no MQMD de uma mensagem acionador gerada pelo gerenciador de filas (continuação)

<b>Campo no MQMD</b>	<b>Valor Usado</b>
<i>Report</i>	MQRO_NONE
<i>MsgType</i>	MQMT_DATAGRAM
<i>Expiry</i>	MQEI_UNLIMITED
<i>Feedback</i>	MQFB_NONE
<i>Encoding</i>	MQENC_NATIVE
<i>CodedCharSetId</i>	Atributo <b>CodedCharSetId</b> do gerenciador de filas
<i>Format</i>	MQFMT_TRIGGER
<i>Priority</i>	Atributo <b>DefPriority</b> da fila de inicialização
<i>Persistence</i>	MQPER_NOT_PERSISTENT
<i>MsgId</i>	Um valor exclusivo
<i>CorrelId</i>	MQCI_NONE
<i>BackoutCount</i>	0
<i>ReplyToQ</i>	Espaços em branco
<i>ReplyToQMGr</i>	Nome do gerenciador de filas
<i>UserIdentifier</i>	Espaços em branco
<i>AccountingToken</i>	MQACT_NONE
<i>AppIdentityData</i>	Espaços em branco
<i>PutAppType</i>	MQAT_QMGR ou conforme apropriado para o agente do canal de mensagens
<i>PutAppName</i>	Primeiros 28 bytes do nome do gerenciador de filas
<i>PutDate</i>	Data quando a mensagem do acionador é enviada
<i>PutTime</i>	Horário quando a mensagem do acionador é enviada
<i>AppOriginData</i>	Espaços em branco

Um aplicativo que gera uma mensagem do acionador é recomendado para configurar valores semelhantes, exceto o seguinte:

- O campo *Priority* pode ser configurado como MQPRI\_PRIORITY\_AS\_Q\_DEF (o gerenciador de filas mudará isso para a prioridade padrão para a fila de inicialização quando a mensagem for colocada).
- O campo *ReplyToQMGr* pode ser definido como espaços em branco (o gerenciador de filas alterará isso para o nome do gerenciador de fila local quando a mensagem for colocada)...
- Configure os campos de contexto conforme apropriado para o aplicativo..

### **StrucId (MQCHAR4) para MQTM**

Este é o identificador da estrutura da mensagem do acionador. Ele é sempre um campo de entrada Seu valor é MQTM\_STRUC\_ID.

O valor deve ser:.

#### **MQTM\_STRUC\_ID**

Identificador para a estrutura da mensagem do acionador

Para a linguagem de programação C, a constante MQTM\_STRUC\_ID\_ARRAY também é definida. Ele tem o mesmo valor que MQTM\_STRUC\_ID, mas é uma matriz de caracteres em vez de uma cadeia.

## ***Versão (MQLONG) para MQTM***

Este é o número da versão da estrutura O valor deve ser:

### **MQTM\_VERSION\_1**

Número da versão para a estrutura da mensagem do acionador

A constante a seguir especifica o número da versão atual:

### **MQTM\_CURRENT\_VERSION**

A versão atual da estrutura da mensagem do acionador

O valor inicial desse campo é MQTM\_VERSION\_1.

## ***QName (MQCHAR48) para MQTM***

Este é o nome da fila para a qual ocorreu um evento acionador e é usado pelo aplicativo iniciado pelo aplicativo do monitor acionador O gerenciador de filas inicializa esse campo com o valor do atributo **QName** da fila acionada; consulte [“Atributos para filas” na página 865](#) para obter detalhes desse atributo.

Os nomes que são mais curtos que o comprimento definido do campo são preenchidos à direita com espaços em branco; eles não são encerrados prematuramente por um caractere nulo.

O comprimento desse campo é fornecido por MQ\_Q\_NAME\_LENGTH O valor inicial deste campo é a cadeia nula em C e 48 caracteres em branco em outras linguagens de programação.

## ***ProcessName (MQCHAR48) para MQTM***

Este é o nome do objeto do processo do gerenciador de filas especificado para a fila acionada e pode ser usado pelo aplicativo do monitor acionador que recebe a mensagem do acionador O gerenciador de filas inicializa esse campo com o valor do atributo **ProcessName** da fila identificada pelo campo *QName* ; consulte [“Atributos para filas” na página 865](#) para obter detalhes desse atributo.

Os nomes que são menores que o comprimento definido do campo são sempre preenchidos à direita com espaços em branco; eles não são terminados prematuramente por um caractere nulo.

O comprimento desse campo é fornecido por MQ\_PROCESS\_NAME\_LENGTH O valor inicial deste campo é a cadeia nula em C e 48 caracteres em branco em outras linguagens de programação.

## ***TriggerData (MQCHAR64) para MQTM***

Esses são dados de formato livre para uso pelo aplicativo de monitor acionador que recebe a mensagem do acionador. O gerenciador de filas inicializa esse campo com o valor do atributo **TriggerData** da fila identificada pelo campo *QName* ; consulte [“Atributos para filas” na página 865](#) para obter detalhes desse atributo. O conteúdo desses dados não é significativo para o gerenciador da fila.

No z/OS, para um aplicativo CICS iniciado usando a transação CKTI, essas informações não são usadas..

O comprimento desse campo é fornecido por MQ\_TRIGGER\_DATA\_LENGTH O valor inicial desse campo é a sequência nula em C e 64 caracteres em branco em outras linguagens de programação.

## ***ApplType (MQLONG) para MQTM***

Isso identifica a natureza do programa a ser iniciado e é usado pelo aplicativo monitor acionador que recebe a mensagem do acionador. O gerenciador de filas inicializa esse campo com o valor do atributo **ApplType** do objeto de processo identificado pelo campo *ProcessName* ; consulte [“Atributos para Definições de Processo” na página 901](#) para obter detalhes desse atributo. O conteúdo desses dados não é significativo para o gerenciador da fila.

*ApplType* pode ter um dos seguintes valores padrão. Os tipos definidos pelo usuário também podem ser usados, mas devem ser restritos a valores no intervalo MQAT\_USER\_FIRST até MQAT\_USER\_LAST:

### **MQAT\_AIX**

AIX (mesmo valor que MQAT\_UNIX).

**MQAT\_BATCH**

aplicativo em lote

**MQAT\_BROKER**

Aplicativo Broker

**MQAT\_CICS**

Transação CICS .

**MQAT\_CICS\_BRIDGE**

CICS bridge .

**MQAT\_CICS\_VSE**

Transação CICS/VSE .

**MQAT\_DOS**

IBM MQ MQI client aplicação no PC DOS.

**MQAT\_IMS**

IMS .

**MQAT\_IMS\_BRIDGE**

aplicativo de ponte IMS .

**MQAT\_JAVA**

Java .

**MQAT\_MVS**

Aplicativo MVS ou TSO (mesmo valor que MQAT\_ZOS).

**MQAT\_NOTES\_AGENT**

Lotus Notes Aplicativo do agente..

**MQAT\_OS390**

OS/390 (mesmo valor que MQAT\_ZOS).

**MQAT\_OS400**

IBM i .

**MQAT\_RRS\_BATCH**

Aplicação em lote do RRS

**MQAT\_UNIX**

UNIX .

**MQAT\_UNKNOWN**

Aplicação de tipo desconhecido..

**MQAT\_USER**

Tipo de aplicativo definido pelo usuário..

**MQAT\_VOS**

Aplicativo do Stratus VOS

**MQAT\_WINDOWS**

Aplicativo Windows de 16 bits

**MQAT\_WINDOWS\_NT**

aplicativo Windows de 32 bits.

**MQAT\_WLM**

Aplicativo do gerenciador de carga de trabalho do z/OS

**MQAT\_XCF**

XCF.

**MQAT\_ZOS**

z/OS .

**MQAT\_USER\_FIRST**

Valor mais baixo para o tipo de aplicativo definido pelo usuário.

**MQAT\_USER\_LAST**

Valor mais alto para o tipo de aplicativo definido pelo usuário.

O valor inicial desse campo é 0.

### ***AppId (MQCHAR256) para MQTM***

Essa é uma sequência de caracteres que identifica o aplicativo a ser iniciado e é usada pelo aplicativo acionador-monitor que recebe a mensagem do acionador O gerenciador de filas inicializa esse campo com o valor do atributo **AppId** do objeto de processo identificado pelo campo *ProcessName* ; consulte [“Atributos para Definições de Processo”](#) na página 901 para obter detalhes desse atributo. O conteúdo desses dados não é significativo para o gerenciador da fila.

O significado de *AppId* é determinado pelo aplicativo do monitor acionador O monitor acionador fornecido pelo IBM MQ requer que *AppId* seja o nome de um programa executável As notas a seguir se aplicam aos ambientes indicados:

- Em z/OS, *AppId* é:
  - Um identificador de transação CICS , para aplicativos iniciados usando a transação CKTI do monitor acionador CICS
  - Um identificador de transação IMS , para aplicativos iniciados usando o IMS monitor acionador CSQQTRMN
- Em sistemas Windows , o nome do programa pode ser prefixado com um caminho de unidade e de diretório
- No IBM i, o nome do programa pode ser prefixado com um nome de biblioteca e / caractere
- No AIX and Linux, o nome do programa pode ser prefixado com um caminho do diretório

O comprimento desse campo é fornecido por MQ\_PROCESS\_APPL\_ID\_LENGTH O valor inicial desse campo é a sequência nula em C e 256 caracteres em branco em outras linguagens de programação.

### ***EnvData (MQCHAR128) para MQTM***

Esta é uma sequência de caracteres que contém informações relacionadas ao ambiente pertencentes ao aplicativo a ser iniciado e é usada pelo aplicativo acionador-monitor que recebe a mensagem do acionador O gerenciador de filas inicializa esse campo com o valor do atributo **EnvData** do objeto de processo identificado pelo campo *ProcessName* ; consulte [“Atributos para Definições de Processo”](#) na página 901 para obter detalhes desse atributo. O conteúdo desses dados não é significativo para o gerenciador da fila.

No z/OS, para um aplicativo CICS iniciado usando a transação CKTI, ou um aplicativo IMS a ser iniciado usando a transação CSQQTRMN, essas informações não são usadas

O comprimento desse campo é fornecido por MQ\_PROCESS\_ENV\_DATA\_LENGTH O valor inicial deste campo é a cadeia nula em C, e 128 caracteres em branco em outras linguagens de programação.

### ***UserData (MQCHAR128) para MQTM***

Essa é uma sequência de caracteres que contém informações do usuário relevantes para o aplicativo a ser iniciado e é usada pelo aplicativo monitor acionador que recebe a mensagem do acionador. O gerenciador de filas inicializa esse campo com o valor do atributo **UserData** do objeto de processo identificado pelo campo *ProcessName* ; consulte [“Atributos para Definições de Processo”](#) na página 901 para obter detalhes desse atributo. O conteúdo desses dados não é significativo para o gerenciador da fila.

Para o Microsoft Windows, a sequência de caractere não deverá conter aspas duplas se a definição de processo for transmitida para **runmqtm**

O comprimento desse campo é fornecido por MQ\_PROCESS\_USER\_DATA\_LENGTH O valor inicial deste campo é a cadeia nula em C, e 128 caracteres em branco em outras linguagens de programação.



## MQTMC2 - Mensagem do acionador 2 (formato de caractere)

Quando um aplicativo de monitor acionador recupera uma mensagem do acionador (MQTM) de uma fila de inicialização, o monitor acionador pode precisar transmitir algumas ou todas as informações na mensagem do acionador para o aplicativo que o monitor acionador inicia.

Informações que o aplicativo iniciado pode precisar incluem *QName*, *TriggerData* e *UserData*. O aplicativo do monitor acionador pode passar a estrutura MQTM diretamente para o aplicativo iniciado ou passar uma estrutura MQTMC2, dependendo do que é permitido pelo ambiente e conveniente para o aplicativo iniciado.

Essa estrutura faz parte da IBM MQ Trigger Monitor Interface (TMI), que é uma das interfaces de estrutura do IBM MQ.

### Conjunto de caracteres e codificação

Os dados de caractere no MQTMC2 estão no conjunto de caractere do gerenciador de filas locais; isso é fornecido pelo atributo do gerenciador de filas **CodedCharSetId** ...

### Uso

A estrutura MQTMC2 é muito semelhante ao formato da estrutura MQTM. A diferença é que os campos sem caractere no MQTM são alterados no MQTMC2 para campos de caracteres do mesmo comprimento e o nome do gerenciador de filas é incluído no final da estrutura.

- ▶ **z/OS** No z/OS, para um aplicativo MQAT\_IMS que é iniciado usando o aplicativo CSQQTRMN, uma estrutura MQTMC2 é disponibilizada para o aplicativo iniciado.
- ▶ **IBM i** No IBM i, o aplicativo de monitor acionador fornecido com IBM MQ passa uma estrutura MQTMC2 para o aplicativo iniciado.

### Campos

**Nota:** Na tabela a seguir, os campos são agrupados por uso e não alfabeticamente. Os tópicos filhos seguem a mesma sequência.

Nome e descrição do campo	Nome da constante	Valor inicial (se houver) da constante
<u>StrucId</u> (identificador de estrutura)	MQTMC_STRUC_ID	' TMC↵ '
<u>Versão</u> (número da versão da estrutura).	MQTMC_VERSION_2	' ↵↵↵2 '
<u>QName</u> (nome da fila acionada)	Nenhum	Cadeia nula ou espaços em branco
<u>ProcessName</u> (nome do objeto de processo)	Nenhum	Cadeia nula ou espaços em branco
<u>TriggerData</u> (dados do acionador)	Nenhum	Cadeia nula ou espaços em branco
<u>ApplType</u> (tipo de aplicativo)	Nenhum	Espaços em branco
<u>AppId</u> (identificador do aplicativo).	Nenhum	Cadeia nula ou espaços em branco
<u>EnvData</u> (dados do ambiente)	Nenhum	Cadeia nula ou espaços em branco

Tabela 535. Campos em MQTMC2 (continuação)

Nome e descrição do campo	Nome da constante	Valor inicial (se houver) da constante
<u>UserData</u> (dados do usuário)	Nenhum	Cadeia nula ou espaços em branco
<u>QMgrName</u> (nome do gerenciador de filas)	Nenhum	Cadeia nula ou espaços em branco

**Notas:**

1. O símbolo ~ representa um único caractere em branco.
2. O valor Sequência nula ou espaços em branco indica a sequência nula em C e caracteres em branco em outras linguagens de programação.
3. Na linguagem de programação C, a variável macroMQTMC2\_DEFAULT contém os valores listados acima.. Use-o da seguinte maneira para fornecer valores iniciais para os campos na estrutura:

```
MQTMC2 MyTMC = {MQTMC2_DEFAULT};
```

## Declarações de idiomas

### Declaração C para MQTMC2

```
typedef struct tagMQTMC2 MQTMC2;
struct tagMQTMC2 {
    MQCHAR4    StrucId;        /* Structure identifier */
    MQCHAR4    Version;       /* Structure version number */
    MQCHAR48   QName;         /* Name of triggered queue */
    MQCHAR48   ProcessName;   /* Name of process object */
    MQCHAR64   TriggerData;   /* Trigger data */
    MQCHAR4    ApplType;      /* Application type */
    MQCHAR256  ApplId;        /* Application identifier */
    MQCHAR128  EnvData;       /* Environment data */
    MQCHAR128  UserData;      /* User data */
    MQCHAR48   QMgrName;     /* Queue manager name */
};
```

### Declaração COBOL para MQTMC2

```
** MQTMC2 structure
10 MQTMC2.
** Structure identifier
15 MQTMC2-STRUCID PIC X(4).
** Structure version number
15 MQTMC2-VERSION PIC X(4).
** Name of triggered queue
15 MQTMC2-QNAME PIC X(48).
** Name of process object
15 MQTMC2-PROCESSNAME PIC X(48).
** Trigger data
15 MQTMC2-TRIGGERDATA PIC X(64).
** Application type
15 MQTMC2-APPLTYPE PIC X(4).
** Application identifier
15 MQTMC2-APPLID PIC X(256).
** Environment data
15 MQTMC2-ENVDATA PIC X(128).
** User data
15 MQTMC2-USERDATA PIC X(128).
** Queue manager name
15 MQTMC2-QMGRNAME PIC X(48).
```

## Declaração de PL/I para MQTMC2

```
dc1
1 MQTMC2 based,
3 StrucId      char(4),    /* Structure identifier */
3 Version      char(4),    /* Structure version number */
3 QName        char(48),   /* Name of triggered queue */
3 ProcessName  char(48),   /* Name of process object */
3 TriggerData  char(64),   /* Trigger data */
3 ApplType     char(4),    /* Application type */
3 ApplId       char(256),  /* Application identifier */
3 EnvData      char(128),  /* Environment data */
3 UserData     char(128),  /* User data */
3 QMgrName     char(48);   /* Queue manager name */
```

## High Level Assembler declaração para MQTMC2

```
MQTMC2          DSECT
MQTMC2_STRUCID  DS   CL4    Structure identifier
MQTMC2_VERSION  DS   CL4    Structure version number
MQTMC2_QNAME    DS   CL48   Name of triggered queue
MQTMC2_PROCESSNAME DS CL48   Name of process object
MQTMC2_TRIGGERDATA DS CL64   Trigger data
MQTMC2_APPLTYPE DS   CL4    Application type
MQTMC2_APPLID   DS   CL256  Application identifier
MQTMC2_ENVDATA  DS   CL128  Environment data
MQTMC2_USERDATA DS   CL128  User data
MQTMC2_QMGRNAME DS   CL48   Queue manager name
*
MQTMC2_LENGTH   EQU   *-MQTMC2
                ORG   MQTMC2
MQTMC2_AREA     DS   CL(MQTMC2_LENGTH)
```

## Declaração do Visual Basic para MQTMC2

```
Type MQTMC2
StrucId      As String*4    'Structure identifier'
Version      As String*4    'Structure version number'
QName        As String*48   'Name of triggered queue'
ProcessName  As String*48   'Name of process object'
TriggerData  As String*64   'Trigger data'
ApplType     As String*4    'Application type'
ApplId       As String*256  'Application identifier'
EnvData      As String*128  'Environment data'
UserData     As String*128  'User data'
QMgrName     As String*48   'Queue manager name'
End Type
```

### **StrucId (MQCHAR4) para MQTMC2**

Este é o identificador da estrutura da mensagem do acionador 2 (formato de caractere). Ele é sempre um campo de entrada Seu valor é MQTMC2\_STRUC\_ID.

O valor deve ser:.

#### **MQTMC2\_STRUC\_ID**

Identificador para a estrutura da mensagem do acionador (formato de caractere)

Para a linguagem de programação C, a constante MQTMC2\_STRUC\_ID\_ARRAY também é definida. Isso tem o mesmo valor que MQTMC2\_STRUC\_ID, mas é uma matriz de caracteres em vez de uma cadeia.

### **Versão (MQCHAR4) para MQTMC2**

Número de versão da estrutura.

O valor deve ser:.

#### **MQTMC\_VERSION\_2**

Estrutura da mensagem do acionador versão 2 (formato de caractere).

Para a linguagem de programação C, a constante MQTMC\_VERSION\_2\_ARRAY também é definida; isso tem o mesmo valor que MQTMC\_VERSION\_2, mas é uma matriz de caracteres em vez de uma cadeia.

A constante a seguir especifica o número da versão atual:

#### **MQTMC\_CURRENT\_VERSION**

Versão atual da estrutura da mensagem do acionador (formato de caractere).

#### **QName (MQCHAR48) para MQTMC2**

Nome da fila acionada..

Consulte o campo *QName* na estrutura do MQTM

#### **ProcessName (MQCHAR48) para MQTMC2 .**

Nome do objeto de processo..

Consulte o campo *ProcessName* na estrutura do MQTM

#### **TriggerData (MQCHAR64) para MQTMC2**

Dados do acionador.

Consulte o campo *TriggerData* na estrutura do MQTM

#### **ApplType (MQCHAR4) para MQTMC2**

Tipo de Aplicativo.

Esse campo sempre contém espaços em branco, independentemente do valor no campo *ApplType* na estrutura MQTM da mensagem do acionador original.

#### **ApplId (MQCHAR256) para MQTMC2**

O identificador do aplicativo.

Consulte o campo *ApplId* na estrutura do MQTM

#### **EnvData (MQCHAR128) para MQTMC2**

Dados do ambiente

Consulte o campo *EnvData* na estrutura do MQTM

#### **UserData (MQCHAR128) para MQTMC2 .**

Dados do usuário..

Consulte o campo *UserData* na estrutura do MQTM

#### **QMgrName (MQCHAR48) para MQTMC2**

Nome do gerenciador de filas.

Este é o nome do gerenciador de filas no qual o evento acionador ocorreu

### **MQWIH - Cabeçalho de informações de trabalho**

Se uma mensagem precisar ser processada pelo z/OS Workload Manager (WLM), a mensagem deverá começar com uma estrutura MQWIH. Essa estrutura descreve as informações que devem estar presentes no início de uma mensagem que deve ser manipulada pelo WLM

### **Disponibilidade**

Todos os sistemas IBM MQ , além de clientes IBM MQ conectados a esses sistemas

## Nome do Formato

MQFMT\_WORK\_INFO\_HEADER.

## Conjunto de caracteres e codificação

Os campos na estrutura MQWIH estão no conjunto de caracteres e na codificação fornecidos pelos campos *CodedCharSetId* e *Encoding* na estrutura do cabeçalho que precede MQWIH ou por esses campos na estrutura MQMD se o MQWIH estiver no início dos dados da mensagem do aplicativo

O conjunto de caracteres deve ser um que tenha caracteres de byte único para os caracteres válidos em nomes de filas.

## Uso

Para qualquer plataforma suportada pelo IBM MQ , é possível criar e transmitir uma mensagem que inclua a estrutura MQWIH, mas apenas um gerenciador de filas do IBM MQ for z/OS pode interagir com o WLM Portanto, para que a mensagem chegue ao WLM a partir de um gerenciador de filas nãoz/OS , sua rede do gerenciador de filas deve incluir pelo menos um gerenciador de filas do z/OS por meio do qual a mensagem possa ser roteada

## Campos

**Nota:** Na tabela a seguir, os campos são agrupados por uso e não alfabeticamente. Os tópicos filhos seguem a mesma sequência.

<i>Tabela 536. Campos em MQWIH</i>		
Nome e descrição do campo	Nome da constante	Valor inicial (se houver) da constante
<u>StrucId</u> (identificador de estrutura)	MQWIH_STRUC_ID	'WIH~'
<u>Versão</u> (número da versão da estrutura).	MQWIH_VERSION_1	1
<u>StrucLength</u> (comprimento da estrutura MQWIH)	MQWIH_LENGTH_1	120
<u>Codificação</u> (codificação numérica de dados que seguem MQWIH)	Nenhum	0
<u>CodedCharSetId</u> (identificador do conjunto de caracteres de dados que segue MQWIH)..	MQCCSI_UNDEFINED	0
<u>Formato</u> (nome do formato de dados após MQWIH)	MQFMT_NONE	Espaços em branco
<u>Sinalizadores</u> (sinalizadores)	MQWIH_NONE	0
<u>ServiceName</u> (nome do serviço)	Nenhum	Espaços em branco
<u>ServiceStep</u> (nome da etapa de serviço);	Nenhum	Espaços em branco
<u>MsgToken</u> (token de mensagem);	MQMTOK_NONE	Nulos
<u>Reservado</u> (reservado).	Nenhum	Espaços em branco
<b>Notas:</b>		
1. O símbolo ~ representa um único caractere em branco.		
2. Na linguagem de programação C, a variável macroMQWIH_DEFAULT contém os valores listados na tabela. Use-o da seguinte maneira para fornecer valores iniciais para os campos na estrutura:		
<pre>MQWIH MyWIH = {MQWIH_DEFAULT};</pre>		

## Declarações de idiomas

### Declaração C para MQWIH

```
typedef struct tagMQWIH MQWIH;
struct tagMQWIH {
    MQCHAR4   StrucId;           /* Structure identifier */
    MQLONG    Version;          /* Structure version number */
    MQLONG    StrucLength;      /* Length of MQWIH structure */
    MQLONG    Encoding;         /* Numeric encoding of data that follows
                                MQWIH */
    MQLONG    CodedCharSetId;   /* Character-set identifier of data that
                                follows MQWIH */
    MQCHAR8   Format;           /* Format name of data that follows
                                MQWIH */
    MQLONG    Flags;           /* Flags */
    MQCHAR32  ServiceName;     /* Service name */
    MQCHAR8   ServiceStep;     /* Service step name */
    MQBYTE16  MsgToken;        /* Message token */
    MQCHAR32  Reserved;        /* Reserved */
};
```

### Declaração COBOL para MQWIH

```
** MQWIH structure
10 MQWIH.
** Structure identifier
15 MQWIH-STRUCID PIC X(4).
** Structure version number
15 MQWIH-VERSION PIC S9(9) BINARY.
** Length of MQWIH structure
15 MQWIH-STRUCLength PIC S9(9) BINARY.
** Numeric encoding of data that follows MQWIH
15 MQWIH-ENCODING PIC S9(9) BINARY.
** Character-set identifier of data that follows MQWIH
15 MQWIH-CODEDCHARSETID PIC S9(9) BINARY.
** Format name of data that follows MQWIH
15 MQWIH-FORMAT PIC X(8).
** Flags
15 MQWIH-FLAGS PIC S9(9) BINARY.
** Service name
15 MQWIH-SERVICENAME PIC X(32).
** Service step name
15 MQWIH-SERVICESTEP PIC X(8).
** Message token
15 MQWIH-MSGTOKEN PIC X(16).
** Reserved
15 MQWIH-RESERVED PIC X(32).
```

### Declaração PL/I para MQWIH

```
dcl
1 MQWIH based,
3 StrucId char(4), /* Structure identifier */
3 Version fixed bin(31), /* Structure version number */
3 StrucLength fixed bin(31), /* Length of MQWIH structure */
3 Encoding fixed bin(31), /* Numeric encoding of data that
                           follows MQWIH */
3 CodedCharSetId fixed bin(31), /* Character-set identifier of data
                                that follows MQWIH */
3 Format char(8), /* Format name of data that follows
                  MQWIH */
3 Flags fixed bin(31), /* Flags */
3 ServiceName char(32), /* Service name */
3 ServiceStep char(8), /* Service step name */
3 MsgToken char(16), /* Message token */
3 Reserved char(32); /* Reserved */
```

### Declaração High Level Assembler para MQWIH

MQWIH	DSECT	
MQWIH_STRUCID	DS CL4	Structure identifier
MQWIH_VERSION	DS F	Structure version number

MQWIH_STRUCLength	DS	F	Length of MQWIH structure
MQWIH_ENCODING	DS	F	Numeric encoding of data that follows MQWIH
* MQWIH_CODEDCHARSETID	DS	F	Character-set identifier of data that follows MQWIH
* MQWIH_FORMAT	DS	CL8	Format name of data that follows MQWIH
MQWIH_FLAGS	DS	F	Flags
MQWIH_SERVICENAME	DS	CL32	Service name
MQWIH_SERVICESTEP	DS	CL8	Service step name
MQWIH_MSGTOKEN	DS	XL16	Message token
MQWIH_RESERVED	DS	CL32	Reserved
* MQWIH_LENGTH	EQU	*-MQWIH	
	ORG	MQWIH	
MQWIH_AREA	DS	CL(MQWIH_LENGTH)	

## Declaração Visual Basic para MQWIH

```

Type MQWIH
  StrucId      As String*4  'Structure identifier'
  Version     As Long      'Structure version number'
  StrucLength As Long      'Length of MQWIH structure'
  Encoding    As Long      'Numeric encoding of data that follows'
  CodedCharSetId As Long  'Character-set identifier of data that'
  Format      As String*8  'Format name of data that follows MQWIH'
  Flags      As Long      'Flags'
  ServiceName As String*32 'Service name'
  ServiceStep As String*8  'Service step name'
  MsgToken   As MQBYTE16  'Message token'
  Reserved   As String*32  'Reserved'
End Type

```

### **StrucId (MQCHAR4) para MQWIH.**

Este é o identificador da estrutura do cabeçalho de informações de trabalho. Ele é sempre um campo de entrada Seu valor é MQWIH\_STRUC\_ID.

O valor deve ser:.

#### **MQWIH\_STRUC\_ID**

Identificador para a estrutura do cabeçalho de informações de trabalho.

Para a linguagem de programação C, a constante MQWIH\_STRUC\_ID\_ARRAY também é definida. Ele tem o mesmo valor que MQWIH\_STRUC\_ID, mas é uma matriz de caracteres em vez de uma sequência.

### **Versão (MQLONG) para MQWIH**

Este é o número da versão da estrutura O valor deve ser:.

#### **MQWIH\_VERSION\_1**

Version-1 estrutura do cabeçalho de informações de trabalho.

A constante a seguir especifica o número da versão atual:

#### **MQWIH\_CURRENT\_VERSION**

Versão atual da estrutura do cabeçalho de informações de trabalho

O valor inicial desse campo é MQWIH\_VERSION\_1.

### **StrucLength (MQLONG) para MQWIH**

É o comprimento da estrutura MQWIH. O valor deve ser:.

#### **MQWIH\_LENGTH\_1**

Comprimento da estrutura do cabeçalho de informações de trabalho version-1 .

A constante a seguir especifica o comprimento da versão atual:

### **MQWIH\_CURRENT\_LENGTH**

Comprimento da versão atual da estrutura do cabeçalho de informações de trabalho

O valor inicial desse campo é MQWIH\_LENGTH\_1.

### **Codificação (MQLONG) para MQWIH**

Isso especifica a codificação numérica dos dados que seguem a estrutura MQWIH; ela não se aplica a dados numéricos na própria estrutura MQWIH

Na chamada MQPUT ou MQPUT1 , o aplicativo deve configurar esse campo para o valor apropriado aos dados.

O valor inicial desse campo é 0.

### **CodedCharSetId (MQLONG) para MQWIH**

Isso especifica o identificador do conjunto de caracteres dos dados que seguem a estrutura MQWIH; ele não se aplica aos dados de caracteres na própria estrutura MQWIH...

Na chamada MQPUT ou MQPUT1 , o aplicativo deve configurar esse campo para o valor apropriado aos dados. É possível usar o seguinte valor especial:

#### **MQCCSI\_INHERIT**

Os dados de caractere nos dados *após* essa estrutura estão no mesmo conjunto de caracteres que essa estrutura

O gerenciador de filas altera esse valor na estrutura enviada na mensagem para o identificador do conjunto de caractere real da estrutura.. Desde que não ocorra nenhum erro, o valor MQCCSI\_INHERIT não será retornado pela chamada MQGET

MQCCSI\_INHERIT não poderá ser usado se o valor do campo *PutApplType* no MQMD for MQAT\_BROKER..

O valor inicial desse campo é MQCCSI\_UNDEFINED.

### **Formato (MQCHAR8) para MQWIH**

Isso especifica o nome do formato dos dados que seguem a estrutura MQWIH

Na chamada MQPUT ou MQPUT1 , o aplicativo deve configurar esse campo para o valor apropriado aos dados. As regras para codificação desse campo são as mesmas que aquelas para o campo *Format* no MQMD.

O comprimento desse campo é fornecido por MQ\_FORMAT\_LENGTH O valor inicial desse campo é MQFMT\_NONE.

### **Sinalizadores (MQLONG) para MQWIH**

O valor deve ser:.

#### **MQWIH\_NONE**

Sem bandeiras.

O valor inicial desse campo é MQWIH\_NONE.

### **ServiceName (MQCHAR32) para MQWIH**

É o nome do serviço que deve processar a mensagem.

O comprimento desse campo é fornecido por MQ\_SERVICE\_NAME\_LENGTH O valor inicial desse campo é 32 caracteres em branco.

### **ServiceStep (MQCHAR8) para MQWIH**

Este é o nome da etapa de *ServiceName* à qual a mensagem se relaciona



O comprimento desse campo é fornecido por MQ\_SERVICE\_STEP\_LENGTH O valor inicial deste campo é 8 caracteres em branco.

### **MsgToken (MQBYTE16) para MQWIH**

Este é um token de mensagem que identifica exclusivamente a mensagem..

Para as chamadas MQPUT e MQPUT1 , esse campo é ignorado. O comprimento desse campo é fornecido por MQ\_MSG\_TOKEN\_LENGTH O valor inicial desse campo é MQMTOK\_NONE.

### **Reservado (MQCHAR32) para MQWIH**

Este é um campo reservado; ele deve estar em branco

## **MQXP-Parâmetro de saída do bloco**

A estrutura MQXP é usada como um parâmetro de entrada / saída para a saída cruzada da API. Para obter mais informações sobre essa saída, consulte [A saída cruzada da API](#).

## **Conjunto de caracteres e codificação**

Os dados de caractere em MQXP estão no conjunto de caracteres do gerenciador de fila local; isso é fornecido pelos atributos do gerenciador de filas do **CodedCharSetId** Os dados numéricos em MQXP estão na codificação da máquina nativa; isso é fornecido por MQENC\_NATIVE

## **Campos**

**Nota:** Na tabela a seguir, os campos são agrupados por uso e não alfabeticamente. Os tópicos filhos seguem a mesma sequência.

<i>Tabela 537. Campos em MQXP</i>	
<b>Nome e descrição do campo</b>	<b>Nome da constante</b>
<u>StrucId</u> (identificador de estrutura)	MQXP_STRUC_ID
<u>Versão</u> (número da versão da estrutura).	MQXP_VERSION_1
<u>ExitId</u> (identificador de saída)	MQXT_API_CROSSING_EXIT
<u>ExitReason</u> (motivo da chamada de saída)	Nenhum
<u>ExitResponse</u> (resposta da saída)	Nenhum
<u>ExitCommand</u> (código de chamada API)	Nenhum
<u>ExitParmCount</u> (contagem de parâmetros)	Nenhum
<u>Reservado</u> (reservado).	Nenhum
<u>ExitUserÁrea</u> (área do usuário)	Nenhum

## **Declarações de idiomas**

Declaração C para MQXP

```
typedef struct tagMQXP MQXP;  
struct tagMQXP {  
    MQCHAR4    StrucId;        /* Structure identifier */  
    MQLONG     Version;       /* Structure version number */
```

```

MQLONG    ExitId;          /* Exit identifier */
MQLONG    ExitReason;     /* Reason for invocation of exit */
MQLONG    ExitResponse;   /* Response from exit */
MQLONG    ExitCommand;    /* API call code */
MQLONG    ExitParmCount;  /* Parameter count */
MQLONG    Reserved;       /* Reserved */
MQBYTE16  ExitUserArea;   /* User area */
};

```

### Declaração COBOL para MQXP

```

**  MQXP structure
  10 MQXP.
**  Structure identifier
  15 MQXP-STRUCID      PIC X(4).
**  Structure version number
  15 MQXP-VERSION     PIC S9(9) BINARY.
**  Exit identifier
  15 MQXP-EXITID      PIC S9(9) BINARY.
**  Reason for invocation of exit
  15 MQXP-EXITREASON  PIC S9(9) BINARY.
**  Response from exit
  15 MQXP-EXITRESPONSE PIC S9(9) BINARY.
**  API call code
  15 MQXP-EXITCOMMAND PIC S9(9) BINARY.
**  Parameter count
  15 MQXP-EXITPARMCOUNT PIC S9(9) BINARY.
**  Reserved
  15 MQXP-RESERVED     PIC S9(9) BINARY.
**  User area
  15 MQXP-EXITUSERAREA PIC X(16).

```

### Declaração PL/I para MQXP

```

dcl
  1 MQXP based,
  3 StrucId      char(4),          /* Structure identifier */
  3 Version      fixed bin(31),   /* Structure version number */
  3 ExitId       fixed bin(31),   /* Exit identifier */
  3 ExitReason   fixed bin(31),   /* Reason for invocation of exit */
  3 ExitResponse fixed bin(31),   /* Response from exit */
  3 ExitCommand  fixed bin(31),   /* API call code */
  3 ExitParmCount fixed bin(31), /* Parameter count */
  3 Reserved     fixed bin(31),   /* Reserved */
  3 ExitUserArea char(16);        /* User area */

```

### Declaração High Level Assembler para MQXP

```

MQXP          DSECT
MQXP_STRUCID  DS  CL4  Structure identifier
MQXP_VERSION  DS  F    Structure version number
MQXP_EXITID   DS  F    Exit identifier
MQXP_EXITREASON DS  F  Reason for invocation of exit
MQXP_EXITRESPONSE DS  F  Response from exit
MQXP_EXITCOMMAND DS  F  API call code
MQXP_EXITPARMCOUNT DS  F  Parameter count
MQXP_RESERVED DS  F    Reserved
MQXP_EXITUSERAREA DS  XL16 User area
*
MQXP_LENGTH   EQU  *-MQXP
               ORG  MQXP
MQXP_AREA     DS  CL(MQXP_LENGTH)

```

### **StrucId (MQCHAR4) para MQXP**

Este é o identificador da estrutura do parâmetro de saída. Ele é sempre um campo de entrada Seu valor é MQXP\_STRUC\_ID.

O valor deve ser:.

#### **MQXP\_STRUC\_ID**

Identificador para a estrutura do parâmetro de saída.

Para a linguagem de programação C, a constante MQXP\_STRUC\_ID\_ARRAY também é definida. Ele tem o mesmo valor que MQXP\_STRUC\_ID, mas é uma matriz de caracteres em vez de uma cadeia.

### ***Versão (MQLONG) para MQXP***

Este é o número da versão da estrutura. O valor deve ser:

#### **MQXP\_VERSION\_1**

Número da versão para a estrutura do bloco de parâmetros de saída

**Nota:** Quando uma nova versão desta estrutura é introduzida, o layout da peça existente não é alterado. Portanto, a saída deve verificar se o número da versão é igual ou maior que a versão mais baixa que contém os campos que a saída precisa usar.

Este é um campo de entrada para a saída

### ***ExitId (MQLONG) para MQXP***

Isso é configurado na entrada para a rotina de saída e indica o tipo de saída:

#### **MQXT\_API\_CROSSING\_EXIT**

Saída de cruzamento de API para CICS

Este é um campo de entrada para a saída

### ***ExitReason (MQLONG) para MQXP***

Isso é configurado na entrada para a rotina de saída. Para a saída cruzada da API, ela indica se a rotina é chamada antes ou após a execução da chamada da API:

#### **MQXR\_BEFORE**

Antes da execução da API

#### **MQXR\_AFTER**

Após a execução da API

Este é um campo de entrada para a saída

### ***ExitResponse (MQLONG) para MQXP***

O valor é configurado pela saída para comunicação com o responsável pela chamada. Os seguintes valores são definidos:

#### **MQXCC\_OK**

Saída concluída com êxito.

#### **MQXCC\_SUPPRESS\_FUNCTION**

Suprimir função.

Quando esse valor é configurado por uma saída de cruzamento de API chamada *antes* da chamada API, a chamada API não é executada. O *CompCode* para a chamada é configurado para MQCC\_FAILED, o *Reason* é configurado como MQRC\_SUPPRESSED\_BY\_EXIT e todos os outros parâmetros permanecem como a saída os deixou.

Quando esse valor é configurado por uma saída de cruzamento de API chamada *após* a chamada de API, ele é ignorado pelo gerenciador de filas.

#### **MQXCC\_SKIP\_FUNCTION**

Ignorar função.

Quando esse valor é configurado por uma saída de cruzamento de API chamada *antes* da chamada de API, a chamada de API não é executada; o *CompCode* e *Reason* e todos os outros parâmetros permanecem como a saída os deixou.

Quando esse valor é configurado por uma saída de cruzamento de API chamada *após* a chamada de API, ele é ignorado pelo gerenciador de filas.

Este é um campo de saída da saída.

### ***ExitCommand (MQLONG) para MQXP***

Este campo é configurado na entrada para a rotina de saída Ele identifica a chamada API que fez com que a saída fosse chamada:

#### **MQXC\_CALLBACK**

A chamada CALLBACK.

#### **MQXC\_MQBACK**

A chamada MQBACK.

#### **MQXC\_MQCB**

A chamada de MQCB

#### **MQXC\_MQCLOSE**

A chamada MQCLOSE.

#### **MQXC\_MQCMIT**

A chamada de MQCMIT

#### **MQXC\_MQCTL**

A chamada de MQCTL

#### **MQXC\_MQGET**

A chamada MQGET..

#### **MQXC\_MQINQ**

A chamada de MQINQ

#### **MQXC\_MQOPEN**

A chamada de MQOPEN

#### **MQXC\_MQPUT**

A chamada MQPUT..

#### **MQXC\_MQPUT1**

A chamada de MQPUT1 .

#### **MQXC\_MQSET**

A chamada de MQSET

#### **MQXC\_MQSTAT**

A chamada de MQSTAT

#### **MQXC\_MQSUB**

A chamada de MQSUB

#### **MQXC\_MQSUBRQ**

A chamada de MQSUBRQ

Este é um campo de entrada para a saída

### ***ExitParmContagem (MQLONG) para MQXP***

Este campo é configurado na entrada para a rotina de saída Ele contém o número de parâmetros que a chamada MQ usa.

*Tabela 538. Número de parâmetros para cada chamada do MQ*

---

<b>Nome da chamada</b>	<b>Número de parâmetros</b>
MQBACK	3
MQCLOSE	5
MQCMIT	3
MQGET	9
MQINQ	10
MQOPEN	6

Tabela 538. Número de parâmetros para cada chamada do MQ (continuação)

Nome da chamada	Número de parâmetros
MQPUT	8
MQPUT1	8
MQSET	10

Este é um campo de entrada para a saída

### **Reservado (MQLONG) para MQXP**

Este é um campo reservado Seu valor não é significativo à saída.

### **Área ExitUser(MQBYTE16) para MQXP**

Este é um campo que está disponível para a saída usar. É inicializado para zero binário para o comprimento do campo antes da primeira chamada da saída para a tarefa e, depois disso, quaisquer mudanças feitas nesse campo pela saída são preservadas nas chamadas da saída.. O seguinte valor é definido:

#### **MQXUA\_NONE**

Nenhuma informação do usuário.

O valor é binário zero para o comprimento do campo.

Para a linguagem de programação C, a constante MQXUA\_NONE\_ARRAY também é definida; isso tem o mesmo valor que MQXUA\_NONE, mas é uma matriz de caracteres em vez de uma cadeia.

O comprimento desse campo é fornecido por MQ\_EXIT\_USER\_AREA\_LENGTH Este é um campo de entrada / saída na saída.

### **MQXQH-Cabeçalho da fila de transmissão**

A estrutura MQXQH descreve as informações prefixadas para os dados da mensagem do aplicativo de mensagens quando elas estão em filas de transmissão. Uma fila de transmissão é um tipo especial de fila local que contém temporariamente mensagens destinadas para filas remotas (ou seja, destinadas para filas que não pertencem ao gerenciador de fila local). Uma fila de transmissão é denotada pelo atributo da fila **Usage** com o valor MQUS\_TRANSMISSION.

#### **Nome do Formato**

MQFMT\_XMIT\_Q\_HEADER

#### **Conjunto de caracteres e codificação**

Os dados em MQXQH devem estar no conjunto de caracteres fornecido pelo atributo do gerenciador de fila do **CodedCharSetId** e pela codificação do gerenciador de fila local fornecido por MQENC\_NATIVE

Configure o conjunto de caracteres e a codificação do MQXQH nos campos *CodedCharSetId* e *Encoding* em:

- O MQMD separado (se a estrutura MQXQH estiver no início dos dados da mensagem) ou
- A estrutura do cabeçalho que precede a estrutura MQXQH (todos os outros casos).

#### **Campos**

**Nota:** Na tabela a seguir, os campos são agrupados por uso e não alfabeticamente. Os tópicos filhos seguem a mesma sequência.

Tabela 539. Campos em MQXQH para MQXQH

Nome e descrição do campo	Nome da constante	Valor inicial (se houver) da constante
StrucId (identificador de estrutura)	MQXQH_STRUC_ID	'XQH-'
Versão (número da versão da estrutura).	MQXQH_VERSION_1	1
RemoteQName (nome da fila de origem)	Nenhum	Cadeia nula ou espaços em branco
RemoteQMgrNome (nome do gerenciador de fila de destino).	Nenhum	Cadeia nula ou espaços em branco
MsgDesc (descritor de mensagem original).	Mesmos nomes e valores que MQMD; consulte Tabela 500 na página 436	-

**Notas:**

1. O símbolo - representa um único caractere em branco.
2. O valor Sequência nula ou espaços em branco indica a sequência nula em C e caracteres em branco em outras linguagens de programação.
3. Na linguagem de programação C, a variável macro MQXQH\_DEFAULT contém os valores listados na tabela. Use-o da seguinte maneira para fornecer valores iniciais para os campos na estrutura:

```
MQXQH MyXQH = {MQXQH_DEFAULT};
```

## Declarações de idiomas

### Declaração C para MQXQH

```
typedef struct tagMQXQH MQXQH;
struct tagMQXQH {
    MQCHAR4   StrucId;           /* Structure identifier */
    MQLONG    Version;          /* Structure version number */
    MQCHAR48  RemoteQName;      /* Name of destination queue */
    MQCHAR48  RemoteQMgrName;   /* Name of destination queue manager */
    MQMD1     MsgDesc;          /* Original message descriptor */
};
```

### Declaração COBOL para MQXQH

```
** MQXQH structure
10 MQXQH.
** Structure identifier
15 MQXQH-STRUCID PIC X(4).
** Structure version number
15 MQXQH-VERSION PIC S9(9) BINARY.
** Name of destination queue
15 MQXQH-REMOTEQNAME PIC X(48).
** Name of destination queue manager
15 MQXQH-REMOTEQMGRNAME PIC X(48).
** Original message descriptor
15 MQXQH-MSGDESC.
** Structure identifier
20 MQXQH-MSGDESC-STRUCID PIC X(4).
** Structure version number
20 MQXQH-MSGDESC-VERSION PIC S9(9) BINARY.
** Report options
20 MQXQH-MSGDESC-REPORT PIC S9(9) BINARY.
** Message type
20 MQXQH-MSGDESC-MSGTYPE PIC S9(9) BINARY.
```

```

**      Expiry time
20 MQXQH-MSGDESC-EXPIRY          PIC S9(9) BINARY.
**      Feedback or reason code
20 MQXQH-MSGDESC-FEEDBACK        PIC S9(9) BINARY.
**      Numeric encoding of message data
20 MQXQH-MSGDESC-ENCODING        PIC S9(9) BINARY.
**      Character set identifier of message data
20 MQXQH-MSGDESC-CODEDCHARSETID  PIC S9(9) BINARY.
**      Format name of message data
20 MQXQH-MSGDESC-FORMAT          PIC X(8).
**      Message priority
20 MQXQH-MSGDESC-PRIORITY        PIC S9(9) BINARY.
**      Message persistence
20 MQXQH-MSGDESC-PERSISTENCE     PIC S9(9) BINARY.
**      Message identifier
20 MQXQH-MSGDESC-MSGID           PIC X(24).
**      Correlation identifier
20 MQXQH-MSGDESC-CORRELID        PIC X(24).
**      Backout counter
20 MQXQH-MSGDESC-BACKOUTCOUNT   PIC S9(9) BINARY.
**      Name of reply-to queue
20 MQXQH-MSGDESC-REPLYTOQ        PIC X(48).
**      Name of reply queue manager
20 MQXQH-MSGDESC-REPLYTOQMGR     PIC X(48).
**      User identifier
20 MQXQH-MSGDESC-USERIDENTIFIER  PIC X(12).
**      Accounting token
20 MQXQH-MSGDESC-ACCOUNTINGTOKEN PIC X(32).
**      Application data relating to identity
20 MQXQH-MSGDESC-APPLIDENTITYDATA PIC X(32).
**      Type of application that put the message
20 MQXQH-MSGDESC-PUTAPPLTYPE     PIC S9(9) BINARY.
**      Name of application that put the message
20 MQXQH-MSGDESC-PUTAPPLNAME     PIC X(28).
**      Date when message was put
20 MQXQH-MSGDESC-PUTDATE         PIC X(8).
**      Time when message was put
20 MQXQH-MSGDESC-PUTTIME         PIC X(8).
**      Application data relating to origin
20 MQXQH-MSGDESC-APPLORIGINDATA  PIC X(4).

```

## Declaração PL/I para MQXQH

```

dcl
  1 MQXQH based,
    3 StrucId          char(4),          /* Structure identifier */
    3 Version          fixed bin(31),    /* Structure version number */
    3 RemoteQName      char(48),         /* Name of destination queue */
    3 RemoteQMgrName   char(48),         /* Name of destination queue
                                          manager */
    3 MsgDesc,
    5 StrucId          char(4),          /* Original message descriptor */
    5 StrucId          char(4),          /* Structure identifier */
    5 Version          fixed bin(31),    /* Structure version number */
    5 Report           fixed bin(31),    /* Report options */
    5 MsgType          fixed bin(31),    /* Message type */
    5 Expiry           fixed bin(31),    /* Expiry time */
    5 Feedback         fixed bin(31),    /* Feedback or reason code */
    5 Encoding         fixed bin(31),    /* Numeric encoding of message
                                          data */
    5 CodedCharSetId   fixed bin(31),    /* Character set identifier of
                                          message data */
    5 Format            char(8),          /* Format name of message data */
    5 Priority          fixed bin(31),    /* Message priority */
    5 Persistence      fixed bin(31),    /* Message persistence */
    5 MsgId            char(24),         /* Message identifier */
    5 CorrelId         char(24),         /* Correlation identifier */
    5 BackoutCount     fixed bin(31),    /* Backout counter */
    5 ReplyToQ         char(48),         /* Name of reply-to queue */
    5 ReplyToQMgr      char(48),         /* Name of reply queue manager */
    5 UserIdentifier   char(12),         /* User identifier */
    5 AccountingToken  char(32),         /* Accounting token */
    5 ApplIdentityData char(32),         /* Application data relating to
                                          identity */
    5 PutApplType      fixed bin(31),    /* Type of application that put the
                                          message */
    5 PutApplName      char(28),         /* Name of application that put the
                                          message */
    5 PutDate          char(8),          /* Date when message was put */
    5 PutTime          char(8),          /* Time when message was put */

```

```
5 ApplOriginData char(4); /* Application data relating to
origin */
```

## Declaração High Level Assembler para MQXQH

```
MQXQH DSECT
MQXQH_STRUCID DS CL4 Structure identifier
MQXQH_VERSION DS F Structure version number
MQXQH_REMOTEQNAME DS CL48 Name of destination queue
MQXQH_REMOTEQMGRNAME DS CL48 Name of destination queue
* manager
MQXQH_MSGDESC DS 0F Force fullword alignment
MQXQH_MSGDESC_STRUCID DS CL4 Structure identifier
MQXQH_MSGDESC_VERSION DS F Structure version number
MQXQH_MSGDESC_REPORT DS F Report options
MQXQH_MSGDESC_MSGTYPE DS F Message type
MQXQH_MSGDESC_EXPIRY DS F Expiry time
MQXQH_MSGDESC_FEEDBACK DS F Feedback or reason code
MQXQH_MSGDESC_ENCODING DS F Numeric encoding of message
* data
MQXQH_MSGDESC_CODEDCHARSETID DS F Character set identifier of
* message data
MQXQH_MSGDESC_FORMAT DS CL8 Format name of message data
MQXQH_MSGDESC_PRIORITY DS F Message priority
MQXQH_MSGDESC_PERSISTENCE DS F Message persistence
MQXQH_MSGDESC_MSGID DS XL24 Message identifier
MQXQH_MSGDESC_CORRELID DS XL24 Correlation identifier
MQXQH_MSGDESC_BACKOUTCOUNT DS F Backout counter
MQXQH_MSGDESC_REPLYTOQ DS CL48 Name of reply-to queue
MQXQH_MSGDESC_REPLYTOQMGR DS CL48 Name of reply queue manager
MQXQH_MSGDESC_USERIDENTIFIER DS CL12 User identifier
MQXQH_MSGDESC_ACCOUNTINGTOKEN DS XL32 Accounting token
MQXQH_MSGDESC_APPLIDENTITYDATA DS CL32 Application data relating to
* identity
MQXQH_MSGDESC_PUTAPPLTYPE DS F Type of application that put
* the message
MQXQH_MSGDESC_PUTAPPLNAME DS CL28 Name of application that put
* the message
MQXQH_MSGDESC_PUTDATE DS CL8 Date when message was put
MQXQH_MSGDESC_PUTTIME DS CL8 Time when message was put
MQXQH_MSGDESC_APPLORIGINDATA DS CL4 Application data relating to
* origin
MQXQH_MSGDESC_LENGTH EQU *-MQXQH_MSGDESC
ORG MQXQH_MSGDESC
MQXQH_MSGDESC_AREA DS CL(MQXQH_MSGDESC_LENGTH)
*
MQXQH_LENGTH EQU *-MQXQH
ORG MQXQH
MQXQH_AREA DS CL(MQXQH_LENGTH)
```

## Declaração do Visual Basic para MQXQH

```
Type MQXQH
  StrucId As String*4 'Structure identifier'
  Version As Long 'Structure version number'
  RemoteQName As String*48 'Name of destination queue'
  RemoteQMgrName As String*48 'Name of destination queue manager'
  MsgDesc As MQMD1 'Original message descriptor'
End Type
```

## Campos no Descritor de Mensagens Separado

Uma mensagem que está em uma fila de transmissão possui *dois* descritores de mensagem:

- Um descritor de mensagens é armazenado separadamente dos dados da mensagem; isso é chamado de *descritor de mensagem separado* e é gerado pelo gerenciador de filas quando a mensagem é colocada na fila de transmissão. Alguns dos campos no descritor de mensagens separado são copiados do descritor de mensagens fornecido pelo aplicativo na chamada MQPUT ou MQPUT1.

O descritor de mensagens separado é aquele retornado para o aplicativo no parâmetro **MsgDesc** da chamada MQGET quando a mensagem é removida da fila de transmissão.



- Um segundo descritor de mensagens é armazenado na estrutura MQXQH como parte dos dados da mensagem; isso é chamado de *descritor de mensagens integrado* e é uma cópia do descritor de mensagens que foi fornecido pelo aplicativo na chamada MQPUT ou MQPUT1 (com variações menores).

O descritor de mensagens integrado é sempre um MQMD version-1 . Se a mensagem colocada pelo aplicativo tiver valores não padrão para um ou mais dos campos version-2 no MQMD, uma estrutura MQMDE seguirá o MQXQH e será seguida pelos dados da mensagem do aplicativo (se houver). O MQMDE é:

- Gerado pelo gerenciador de filas (se o aplicativo usar um MQMD version-2 para colocar a mensagem) ou
- Já presente no início dos dados da mensagem do aplicativo (se o aplicativo usar um MQMD version-1 para colocar a mensagem).

O descritor de mensagens integrado é aquele retornado para o aplicativo no parâmetro **MsgDesc** da chamada MQGET quando a mensagem é removida da fila de destino final.

Os campos no descritor de mensagens separado são configurados pelo gerenciador de filas conforme mostrado. Se o gerenciador de filas não suportar o MQMD version-2 , um MQMD version-1 será usado sem perda de função.

Tabela 540. Valores usados para campos no MQMD separado

<b>Campo em MQMD separado.</b>	<b>Valor Usado</b>
<i>StrucId</i>	MQMD_STRUC_ID
<i>Version</i>	MQMD_VERSION_2
<i>Report</i>	Copiado do descritor de mensagens integrado, mas com os bits identificados por MQRO_ACCEPT_UNSUP_IF_XMIT_MASK configurado como zero. (Isso impede que uma mensagem de relatório COA ou COD seja gerada quando uma mensagem é colocada ou removida de uma fila de transmissão.)
<i>MsgType</i>	Copiado do descritor de mensagens integrado.
<i>Expiry</i>	Copiado do descritor de mensagens integrado.
<i>Feedback</i>	Copiado do descritor de mensagens integrado.
<i>Encoding</i>	MQENC_NATIVE (ver nota)
<i>CodedCharSetId</i>	atributo <b>CodedCharSetId</b> do gerenciador de filas.
<i>Format</i>	MQFMT_XMIT_Q_HEADER
<i>Priority</i>	Copiado do descritor de mensagens integrado.
<i>Persistence</i>	Copiado do descritor de mensagens integrado.
<i>MsgId</i>	Um novo valor é gerado pelo gerenciador de filas.. Este identificador de mensagem é diferente do <i>MsgId</i> que o gerenciador de filas pode ter gerado para o descritor de mensagens integrado descrito anteriormente..
<i>CorrelId</i>	O <i>MsgId</i> do descritor de mensagens integrado. Para mensagens sendo colocadas no SYSTEM.CLUSTER.TRANSMIT.QUEUE, <i>CorrelId</i> é reservado para uso interno
<i>BackoutCount</i>	0
<i>ReplyToQ</i>	Copiado do descritor de mensagens integrado.
<i>ReplyToQMGr</i>	Copiado do descritor de mensagens integrado.
<i>UserIdentifier</i>	Copiado do descritor de mensagens integrado.

Tabela 540. Valores usados para campos no MQMD separado (continuação)

<b>Campo em MQMD separado.</b>	<b>Valor Usado</b>
<i>AccountingToken</i>	Copiado do descritor de mensagens integrado. Para mensagens sendo colocadas no SYSTEM.CLUSTER.TRANSMIT.QUEUE, <i>AccountingToken</i> é reservado para uso interno
<i>ApplIdentityData</i>	Copiado do descritor de mensagens integrado.
<i>PutApplType</i>	MQAT_QMGR
<i>PutApplName</i>	Os primeiros 28 bytes do nome do gerenciador de filas
<i>PutDate</i>	Data quando a mensagem foi colocada na fila de transmissão.
<i>PutTime</i>	O horário em que a mensagem foi colocada na fila de transmissão
<i>ApplOriginData</i>	Espaços em branco
<i>GroupId</i>	MQGI_NONE
<i>MsgSeqNumber</i>	1
<i>Offset</i>	0
<i>MsgFlags</i>	MQMF_NONE
<i>OriginalLength</i>	MQOL_UNDEFINED

- No Windows, o valor de MQENC\_NATIVE para Micro Focus COBOL difere do valor para C. O valor no campo *Encoding* no descritor de mensagens separado é sempre o valor para C nesses ambientes; esse valor é 546 em decimal. Além disso, os campos de número inteiro na estrutura MQXQH estão na codificação que corresponde a esse valor (a codificação nativa Intel).

### Campos no descritor de mensagens integradas

Os campos no descritor de mensagens integrado têm os mesmos valores que aqueles no parâmetro **MsgDesc** da chamada MQPUT ou MQPUT1, exceto o seguinte:

- O campo *Version* sempre tem o valor MQMD\_VERSION\_1.
- Se o campo *Priority* tiver o valor MQPRI\_PRIORITY\_AS\_Q\_DEF, ele será substituído pelo valor do atributo **DefPriority** da fila.
- Se o campo *Persistence* tiver o valor MQPER\_PERSISTENCE\_AS\_Q\_DEF, ele será substituído pelo valor do atributo **DefPersistence** da fila
- Se o campo *MsgId* tiver o valor MQMI\_NONE ou a opção MQPMO\_NEW\_MSG\_ID foi especificada ou a mensagem for uma mensagem da lista de distribuição, *MsgId* será substituído por um novo identificador de mensagem gerado pelo gerenciador de filas.

Quando uma mensagem da lista de distribuição é dividida em mensagens da lista de distribuição menores colocadas em filas de transmissão diferentes, o campo *MsgId* em cada um dos novos descritores de mensagens integrados é o mesmo que na mensagem da lista de distribuição original.

- Se a opção MQPMO\_NEW\_CORREL\_ID foi especificada, *CorrelId* será substituído por um novo identificador de correlação gerado pelo gerenciador de filas.
- Os campos de contexto são configurados conforme indicado pelas opções MQPMO\_\*\_CONTEXT especificadas no parâmetro **PutMsgOpts**; os campos de contexto são:

- *AccountingToken*
- *ApplIdentityData*
- *ApplOriginData*
- *PutApplName*
- *PutApplType*

- *PutDate*
- *PutTime*
- *UserIdentifier*
- Os campos version-2 (se estavam presentes) são removidos do MQMD e movidos para uma estrutura MQMDE, se um ou mais dos campos version-2 tiverem um valor não padrão.

## Colocando mensagens em filas remotas

Quando um aplicativo coloca uma mensagem em uma fila remota (especificando o nome da fila remota diretamente ou usando uma definição local da fila remota), o gerenciador de filas locais:

- Cria uma estrutura MQXQH contendo o descritor de mensagens integrado
- Anexa um MQMDE se um for necessário e ainda não estiver presente
- Anexa os dados da mensagem do aplicativo..
- Coloca a mensagem em uma fila de transmissão apropriada

## Colocando mensagens diretamente em filas de transmissão

Um aplicativo também pode colocar uma mensagem diretamente em uma fila de transmissão. Nesse caso, o aplicativo deve prefixar os dados da mensagem do aplicativo com uma estrutura MQXQH, e inicializar os campos com valores apropriados. Além disso, o campo *Format* no parâmetro **MsgDesc** da chamada MQPUT ou MQPUT1 deve ter o valor MQFMT\_XMIT\_Q\_HEADER.

Os dados de caracteres na estrutura MQXQH criada pelo aplicativo devem estar no conjunto de caracteres do gerenciador de filas locais (definido pelo atributo do gerenciador de filas **CodedCharSetId**) e os dados de número inteiro devem estar na codificação de máquina nativa... Além disso, os dados de caractere na estrutura MQXQH devem ser preenchidos com espaços em branco para o comprimento definido do campo; os dados não devem ser encerrados prematuramente usando um caractere nulo, porque o gerenciador de fila não converte os caracteres nulos e subsequentes em espaços em branco na estrutura MQXQH.

Entretanto, o gerenciador de fila não verifica se uma estrutura MQXQH está presente ou se valores válidos foram especificados para os campos.

Os aplicativos não devem colocar suas mensagens diretamente no sistema SYSTEM.CLUSTER.TRANSMIT.QUEUE.

## Obtendo mensagens de filas de transmissão

Os aplicativos que obtêm mensagens de uma fila de transmissão devem processar as informações na estrutura MQXQH de uma maneira apropriada. A presença da estrutura MQXQH no início dos dados da mensagem do aplicativo é indicada pelo valor MQFMT\_XMIT\_Q\_HEADER sendo retornado no campo *Format* no parâmetro **MsgDesc** da chamada MQGET. Os valores retornados nos campos *CodedCharSetId* e *Encoding* no parâmetro **MsgDesc** indicam o conjunto de caracteres e a codificação dos dados de caractere e número inteiro na estrutura MQXQH. O conjunto de caractere e a codificação dos dados da mensagem do aplicativo são definidos nos campos *CodedCharSetId* e *Encoding* no descritor de mensagens integrado.

### **StrucId (MQCHAR4) para MQXQH**

Este é o identificador da estrutura do cabeçalho da fila de transmissão. Ele é sempre um campo de entrada. Seu valor é MQXQH\_STRUC\_ID.

O valor deve ser::

#### **MQXQH\_STRUC\_ID**

Identificador para a estrutura do cabeçalho da fila de transmissão.

Para a linguagem de programação C, a constante MQXQH\_STRUC\_ID\_ARRAY também é definida. Ele tem o mesmo valor que MQXQH\_STRUC\_ID, mas é uma matriz de caracteres em vez de uma cadeia.

## ***Versão (MQLONG) para MQXQH***

Este é o número da versão da estrutura. O valor deve ser:

### **MQXQH\_VERSION\_1**

Número da versão para a estrutura do cabeçalho da fila de transmissão

A constante a seguir especifica o número da versão atual:

### **MQXQH\_CURRENT\_VERSION**

Versão atual da estrutura do cabeçalho da fila de transmissão..

O valor inicial desse campo é MQXQH\_VERSION\_1.

## ***RemoteQName (MQCHAR48) para MQXQH***

Este é o nome da fila de mensagens que é o destino eventual aparente para a mensagem (isso pode provar que não será o destino eventual se, por exemplo, essa fila for definida em *RemoteQMgrName* para ser uma definição local de outra fila remota)

Se a mensagem for uma mensagem de lista de distribuição (ou seja, o campo *Format* no descritor de mensagem integrado é MQFMT\_DIST\_HEADER), *RemoteQName* ficará em branco.

O comprimento desse campo é fornecido por MQ\_Q\_NAME\_LENGTH. O valor inicial deste campo é a cadeia nula em C e 48 caracteres em branco em outras linguagens de programação.

## ***RemoteQMgrNome (MQCHAR48) para MQXQH***

Este é o nome do gerenciador de filas ou do grupo de filas compartilhadas que possui a fila que é o destino eventual aparente para a mensagem

Se a mensagem for uma mensagem da lista de distribuições, *RemoteQMgrName* estará em branco

O comprimento desse campo é fornecido por MQ\_Q\_MGR\_NAME\_LENGTH. O valor inicial deste campo é a cadeia nula em C e 48 caracteres em branco em outras linguagens de programação.

## ***MsgDesc (MQMD1) para MQXQH***

Esse é o descritor de mensagens integrado e é uma cópia próxima do descritor de mensagens MQMD que foi especificado como o parâmetro **MsgDesc** na chamada MQPUT ou MQPUT1 quando a mensagem foi originalmente colocada na fila remota.

**Nota:** Este é um MQMD version-1 .

Os valores iniciais dos campos nessa estrutura são os mesmos da estrutura MQMD.

## **Chamadas de função**

Esta seção fornece informações sobre todas as chamadas MQI possíveis. Descrições, sintaxe, informações de parâmetros, notas de uso e chamadas de idioma para cada idioma possível são fornecidas para cada uma das diferentes chamadas.

### **Referências relacionadas**

 [Exemplos de Saída CEDF de Chamadas MQI](#)

### **Descrições de chamada**

Esta seção descreve chamadas MQI.

- [“MQBACK-Mudanças de retorno” na página 651](#)
- [“MQBEGIN-Iniciar unidade de trabalho” na página 655](#)
- [“MQBUFMH - Converter buffer em identificador de mensagens” na página 658](#)
- [“MQCB-Gerenciar retorno de chamada” na página 662](#)
- [“MQCB\\_FUNCTION-Função de Retorno de Chamada” na página 672](#)

- [“MQCLOSE-Fechar objeto” na página 673](#)
- [“MQCMIT-Confirmar mudanças” na página 681](#)
- [“MQCONN - Conectar o gerenciador de filas” na página 685](#)
- [“MQCONNX - Conectar Gerenciador de Filas \(Estendido\)” na página 693](#)
- [“MQCRTMH-Criar identificador de mensagens” na página 699](#)
- [“MQCTL-Retornos de chamada de controle” na página 702](#)
- [“MQDISC-Desconectar gerenciador de filas” na página 709](#)
- [“MQDLTMH-Excluir identificador de mensagem” na página 713](#)
- [“MQDLTMP-Excluir propriedade de mensagem” na página 715](#)
- [“MQGET - Obter Mensagem” na página 718](#)
- [“MQINQ-Consultar atributos do objeto.” na página 730](#)
- [“MQINQMP-Propriedade da mensagem de consulta” na página 749](#)
- [“MQMHBUF-Converter identificador de mensagem em buffer” na página 755](#)
- [“MQOPEN-Abrir objeto” na página 759](#)
- [“MQPUT-Mensagem de colocação” na página 777](#)
- [“MQPUT1 -Colocar uma mensagem” na página 791](#)
- [“MQSET - configurar atributos do objeto” na página 802](#)
- [“MQSETMP-Configurar propriedade de mensagem” na página 808](#)
- [“MQSTAT-Recuperar informações de status” na página 812](#)
- [“MQMHBUF-Converter identificador de mensagem em buffer” na página 755](#)
- [“MQSUB - Assinatura do registro” na página 816](#)
- [“MQSUBRQ-Pedido de assinatura” na página 824](#)

A ajuda online nas plataformas UNIX , na forma de páginas *man* , está disponível para essas chamadas

**Nota:** As chamadas associadas com a conversão de dados, MQXCNV e MQ\_DATA\_CONV\_EXIT, estão em [“saída de conversão de dados” na página 939](#)

### **Convenções usadas nas descrições de chamada**

Para cada chamada, essa coleção de tópicos fornece uma descrição dos parâmetros e do uso da chamada em um formato independente da linguagem de programação. Isso é seguido por chamadas típicas da chamada e declarações típicas de seus parâmetros em cada uma das linguagens de programação suportadas.

**Importante:** Ao codificar chamadas API do IBM MQ , deve-se assegurar que todos os parâmetros relevantes (conforme descrito nas seções a seguir) sejam fornecidos. A falha em fazer isso pode produzir resultados imprevisíveis.

A descrição de cada chamada contém as seguintes seções:

#### **Nome da chamada**

O nome da chamada, seguido por uma breve descrição do propósito da chamada

#### **Parâmetros**

Para cada parâmetro, o nome é seguido por seu tipo de dados entre parênteses () e um dos seguintes:

##### **entrada**

Você fornece informações no parâmetro quando você faz a chamada

##### **saída**

O gerenciador de filas retorna informações no parâmetro quando a chamada é concluída ou falha..

##### **entrada/saída**

Você fornece informações no parâmetro quando faz a chamada e o gerenciador de filas altera as informações quando a chamada é concluída ou falha.

Por exemplo:

*Compcode* (MQLONG)-saída

Em alguns casos, o tipo de dados é uma estrutura. Em todos os casos, há mais informações sobre o tipo ou a estrutura de dados em [“Tipos de dados elementares”](#) na página 237

Os dois últimos parâmetros em cada chamada são um código de conclusão e um código de razão. O código de conclusão indica se a chamada foi concluída com êxito, parcialmente ou não. Informações adicionais sobre o sucesso parcial ou falha da chamada são fornecidas no código de razão. Para obter mais informações sobre cada conclusão e código de razão, consulte [“Códigos de retorno”](#) na página 905..

### **Observações de Uso**

Informações adicionais sobre a chamada, descrevendo como usá-la e quaisquer restrições sobre seu uso.

### **Chamada de linguagem assembler**

Chamada típica da chamada e declaração de seus parâmetros na linguagem assembler.

### **Chamada C**

Chamada típica da chamada e declaração de seus parâmetros, em C.

### **Chamada COBOL**

Chamada típica da chamada e declaração de seus parâmetros em COBOL.

### **Chamada PL/I**

Chamada típica da chamada e declaração de seus parâmetros, em PL/I.

Todos os parâmetros são transmitidos por referência

### **Chamada do Visual Basic**

Chamada típica da chamada e declaração de seus parâmetros, no Visual Basic.

Outras convenções de notação são:

### **Constantes**

Nomes de constantes são mostrados em maiúsculas; por exemplo, MQOO\_OUTPUT. Um conjunto de constantes com o mesmo prefixo é mostrado da seguinte forma: MQIA\_\*. Consulte [“Constantes”](#) na página 61 para obter o valor de uma constante.

### **Matrizes**

Em algumas chamadas, os parâmetros são matrizes de sequências de caracteres que não possuem tamanhos fixos. Nas descrições desses parâmetros, um n minúsculo representa uma constante numérica. Ao codificar a declaração para esse parâmetro, substitua o n pelo valor numérico necessário.

## ***Usando as chamadas na linguagem C***

Parâmetros que são *somente entrada* e do tipo MQHCONN, MQHOBJ, MQHMSG ou MQLONG são transmitidos por valor. Para todos os outros parâmetros, o *address* do parâmetro é transmitido por valor..

Não é necessário especificar todos os parâmetros que são transmitidos por endereço toda vez que você chamar uma função... Onde você não precisar de um parâmetro específico, especifique um ponteiro nulo como o parâmetro na chamada de função, no lugar do endereço de dados do parâmetro... Parâmetros para os quais isso é possível estão identificados nas descrições de chamada.

Nenhum parâmetro é retornado como o valor da chamada; na terminologia C, isso significa que todas as chamadas retornam void.

### *Declarando o parâmetro Buffer*

Cada uma das chamadas **MQGET**, **MQPUT** e **MQPUT1** possui um parâmetro que possui um tipo de dados indefinido: o parâmetro *Buffer*. Use esse parâmetro para enviar e receber os dados da mensagem do aplicativo

Parâmetros desse tipo são mostrados nos exemplos de C como matrizes de MQBYTE. É possível declarar os parâmetros desta maneira, mas geralmente é mais conveniente declará-los como a estrutura específica que descreve o layout dos dados na mensagem. O protótipo de função declara o parâmetro como um ponteiro para vazio, para que seja possível especificar o endereço de qualquer tipo de dado como o parâmetro na chamada de chamada.

Ponteiro para vazio é um ponteiro para dados de formato indefinido. Ele é definido como:

```
typedef void *PMQVOID;
```

## MQBACK-Mudanças de retorno

A chamada MQBACK indica para o gerenciador de fila que todas as mensagens obtêm e colocam que ocorreram desde o último ponto de sincronização devem ser restauradas.

Mensagens colocadas como parte de uma unidade de trabalho são excluídas; mensagens recuperadas como parte de uma unidade de trabalho são restabelecidas na fila.

- No z/OS, essa chamada é utilizada apenas por programas em lote (incluindo programas DL/I em lote IMS).

## Sintaxe

MQBACK (*Hconn*, *Compcode*, *Motivo*)

## Parâmetros

### Hconn

Tipo: MQHCONN - entrada

Esta manipulação representa a conexão ao gerenciador de filas. O valor de *Hconn* foi retornado por uma chamada anterior MQCONN ou MQCONNX.

### CompCode

Tipo: MQLONG - saída

O código de conclusão; é um dos seguintes:

#### MQCC\_OK

Indica conclusão bem-sucedida.

#### MQCC\_WARNING

Aviso (conclusão parcial).

#### MQCC\_FAILED

A chamada falhou.

### Razão

Tipo: MQLONG - saída

Se *CompCode* for MQCC\_OK:

#### MQRC\_NONE

(0, X'000') Nenhuma razão a relatar.

Se *CompCode* for MQCC\_WARNING:

#### MQRC\_OUTCOME\_PENDING

(2124, X'84C') O resultado da operação de retorno está pendente.

Se *CompCode* for MQCC\_FAILED:

#### MQRC\_ADAPTER\_SERV\_LOAD\_ERROR

(2130, X'852') Não foi possível carregar o módulo de serviço do adaptador.

**MQRC\_API\_EXIT\_ERROR**

(2374, X'946') Falha na saída de API.

**MQRC\_ASID\_MISMATCH**

(2157, X'86D') ASIDs de início e primárias são diferentes.

**MQRC\_CALL\_IN\_PROGRESS**

(2219, X'8AB') Chamada MQI inserida antes da chamada anterior concluída.

**MQRC\_CF\_STRUC\_IN\_USE**

(2346, X'92A') Estrutura de recurso de acoplamento em uso.

**MQRC\_CONNECTION\_BROKEN**

(2009, X'7D9') Conexão ao gerenciador de filas perdida.

**MQRC\_ENVIRONMENT\_ERROR**

(2012, X'7DC') Chamada não válida no ambiente.

**MQRC\_HCONN\_ERROR**

(2018, X'7E2') Manipulação de conexões não válida.

**MQRC\_OBJECT\_DAMAGED**

(2101, X'835') Objeto danificado.

**MQRC\_OUTCOME\_MIXED**

(2123, X'84B') O resultado da operação de confirmação ou de retorno é combinado.

**MQRC\_Q\_MGR\_STOPPING**

(2162, X'872') Gerenciador de filas sendo encerrado.

**MQRC\_RESOURCE\_PROBLEM**

(2102, X'836') Recursos insuficientes do sistema disponíveis.

**MQRC\_STORAGE\_MEDIUM\_FULL**

(2192, X'890') O meio de armazenamento externo está cheio.

**MQRC\_STORAGE\_NOT\_AVAILABLE**

(2071, X'817') Armazenamento insuficiente disponível.

**MQRC\_UNEXPECTED\_ERROR**

(2195, X'893') Ocorreu um erro inesperado.

Para obter informações detalhadas sobre esses códigos, consulte [Mensagens e códigos de razão](#)

## Observações de Uso

1. É possível usar essa chamada apenas quando o próprio gerenciador de filas coordena a unidade de trabalho. Ele pode ser:
  - Uma unidade de trabalho local, em que as mudanças afetam apenas os recursos do MQ
  - Uma unidade global de trabalho, em que as mudanças podem afetar recursos pertencentes a outros gerenciadores de recursos, bem como afetar recursos do MQ .

Para obter detalhes adicionais sobre as unidades de trabalho locais e globais, consulte [“MQBEGIN- Iniciar unidade de trabalho”](#) na página 655
2. Em ambientes nos quais o gerenciador de filas não coordena a unidade de trabalho, use a chamada de retorno apropriada, em vez de MQBACK. O ambiente também pode suportar um retorno implícito causado pela finalização anormal do aplicativo.
  - No z/OS, use as seguintes chamadas:
    - Programas em lote (incluindo IMS programas DL/I em lote) podem usar a chamada MQBACK se a unidade de trabalho afetar apenas recursos do MQ . No entanto, se a unidade de trabalho afetar os recursos do MQ e os recursos pertencentes a outros gerenciadores de recursos (por exemplo, Db2 ), use a chamada SRRBACK fornecida pelo z/OS Recoverable Resource Service (RRS). A chamada SRRBACK retorna mudanças em recursos pertencentes aos gerenciadores de recursos que foram ativados para coordenação RRS.



- Aplicativos CICS devem usar o comando EXEC CICS SYNCPOINT ROLLBACK para voltar a unidade de trabalho. Não use a chamada MQBACK para os aplicativos CICS
  - Os aplicativos IMS (que não sejam programas DL/I em lote) devem usar chamadas IMS como ROLB para voltar a unidade de trabalho. Não use a chamada MQBACK para aplicativos IMS (exceto programas DL/I em lote).
- No IBM i, use essa chamada para unidades locais de trabalho coordenadas pelo gerenciador de fila.. Isso significa que uma definição de confirmação não deve existir no nível da tarefa, ou seja, o comando STRCMTCTL com o parâmetro **CMTSCOPE(\*JOB)** não deve ter sido emitido para a tarefa
3. Se um aplicativo terminar com mudanças não confirmados em uma unidade de trabalho, a disposição dessas mudanças dependerá se o aplicativo será encerrado normalmente ou de forma anormal. Consulte as notas de uso em [“MQDISC-Desconectar gerenciador de filas”](#) na página 709 para obter detalhes adicionais
  4. Quando um aplicativo coloca ou obtém mensagens em grupos ou segmentos de mensagens lógicas, o gerenciador de filas retém informações relacionadas ao grupo de mensagens e mensagens lógicas para as últimas chamadas MQPUT e MQGET bem-sucedidas. Essas informações estão associadas ao identificador de filas e incluem itens como:
    - Os valores dos campos *GroupId, MsgSeqNumber, Offsete MsgFlags* em MQMD..
    - Indica se a mensagem faz parte de uma unidade de trabalho
    - Para a chamada MQPUT: se a mensagem é persistente ou não persistente.

O gerenciador de filas mantém *três* conjuntos de informações de grupo e segmento, um conjunto para cada um dos seguintes:

    - A última chamada MQPUT bem-sucedida (isso pode fazer parte de uma unidade de trabalho)
    - A última chamada MQGET bem-sucedida que removeu uma mensagem da fila (isso pode fazer parte de uma unidade de trabalho).
    - A última chamada MQGET bem-sucedida que navegou em uma mensagem na fila (isso não pode fazer parte de uma unidade de trabalho)
  5. As informações associadas à chamada MQGET são restauradas para o valor que ela tinha antes da primeira chamada MQGET bem-sucedida para essa manipulação de fila na unidade de trabalho atual...
 

As filas que foram atualizadas pelo aplicativo após a unidade de trabalho ter iniciado, mas fora do escopo da unidade de trabalho, não terão suas informações de grupo e de segmento restauradas se a unidade de trabalho for restaurada.

A restauração das informações do grupo e do segmento para seu valor anterior quando uma unidade de trabalho é restaurada permite que o aplicativo difunda um grupo de mensagens grandes ou uma mensagem lógica grande que consiste em muitos segmentos em várias unidades de trabalho e reinicie no ponto correto no grupo de mensagens ou mensagem lógica se uma das unidades de trabalho falhar.

Usar várias unidades de trabalho pode ser vantajoso se o gerenciador de fila local tiver apenas armazenamento de fila limitado. No entanto, o aplicativo deve manter informações suficientes para poder reiniciar a colocação ou obtenção de mensagens no ponto correto se ocorrer uma falha do sistema.

Para obter detalhes de como reiniciar no ponto correto após uma falha do sistema, consulte a opção MQPMO\_LOGICAL\_ORDER descrita em [“MQPMO-Opções de mensagem de colocação”](#) na página 518 e a opção MQGMO\_LOGICAL\_ORDER descrita em [“MQGMO-Obter opções de mensagem”](#) na página 379..

As notas de uso restantes aplicam-se apenas quando o gerenciador de filas coordena as unidades de trabalho
  6. Uma unidade de trabalho tem o mesmo escopo que uma manipulação de conexões.. Todas as chamadas do MQ que afetam uma unidade de trabalho específica devem ser executadas usando o mesmo identificador da conexão. Chamadas emitidas usando um identificador de conexão diferente (por exemplo, chamadas emitidas por outro aplicativo) afetam uma unidade de trabalho diferente.

Consulte o parâmetro **Hconn** descrito em [“MQCONN - Conectar o gerenciador de filas”](#) na página 685 para obter informações sobre o escopo de manipulações de conexão

7. Apenas as mensagens que foram colocadas ou recuperadas como parte da unidade de trabalho atual são afetada por esta chamada
8. Um aplicativo de longa execução que emite chamadas MQGET ou MQPUT ou MQPUT1 dentro de uma unidade de trabalho, mas que nunca emite uma confirmação ou chamada de restauração, pode preencher filas com mensagens que não estão disponíveis para outros aplicativos. Para proteger contra essa possibilidade, o administrador deve configurar o atributo do gerenciador de filas **MaxUncommittedMsgs** para um valor que seja baixo o suficiente para evitar que os aplicativos runaway preencham as filas, mas alto o suficiente para permitir que os aplicativos do sistema de mensagens esperados funcionem corretamente.

## Chamada C

```
MQBACK (Hconn, &CompCode, &Reason);
```

Declare os parâmetros conforme a seguir:

```
MQHCONN  Hconn;      /* Connection handle */
MQLONG   CompCode;   /* Completion code */
MQLONG   Reason;     /* Reason code qualifying CompCode */
```

## Chamada COBOL

```
CALL 'MQBACK' USING HCONN, COMPCODE, REASON.
```

Declare os parâmetros conforme a seguir:

```
** Connection handle
01 HCONN      PIC S9(9) BINARY.
** Completion code
01 COMPCODE   PIC S9(9) BINARY.
** Reason code qualifying COMPCODE
01 REASON     PIC S9(9) BINARY.
```

## Chamada PL/I

```
call MQBACK (Hconn, CompCode, Reason);
```

Declare os parâmetros conforme a seguir:

```
dcl Hconn      fixed bin(31); /* Connection handle */
dcl CompCode   fixed bin(31); /* Completion code */
dcl Reason     fixed bin(31); /* Reason code qualifying CompCode */
```

## Chamada do Assembler de Alto Nível

```
CALL MQBACK,(HCONN,COMPCODE,REASON)
```

Declare os parâmetros conforme a seguir:

```
HCONN      DS F Connection handle
```

```
COMPCODE DS F Completion code
REASON   DS F Reason code qualifying COMPCODE
```

## Chamada do Visual Basic

```
MQBACK Hconn, CompCode, Reason
```

Declare os parâmetros conforme a seguir:

```
Dim Hconn As Long 'Connection handle'
Dim CompCode As Long 'Completion code'
Dim Reason As Long 'Reason code qualifying CompCode'
```

## MQBEGIN-Iniciar unidade de trabalho

A chamada MQBEGIN inicia uma unidade de trabalho coordenada pelo gerenciador de filas e que pode envolver gerenciadores de recursos externos.

### Sintaxe

MQBEGIN (*Hconn*, *BeginOptions*, *Compcode*, *Razão*)

### Parâmetros

#### Hconn

Tipo: MQHCONN - entrada

Esta manipulação representa a conexão ao gerenciador de filas. O valor de *Hconn* foi retornado por uma chamada anterior MQCONN ou MQCONNX.

*Hconn* deve ser uma manipulação de conexões não compartilhadas. Se um identificador de conexão compartilhada for especificado, a chamada falhará com o código de razão MQRC\_HCONN\_ERROR. Consulte a descrição das opções MQCNO\_HANDLE\_SHARE\_\* em [“MQCNO-Opções de conexão”](#) na página 324 para obter mais informações sobre manipulações compartilhadas ou não compartilhadas...

#### BeginOptions

Tipo: MQBO-entrada/saída

Estas são as opções que controlam a ação de MQBEGIN, conforme descrito em [“MQBO-Opções de início de”](#) na página 285.

Se nenhuma opção for necessária, os programas gravados no assembler C ou S/390 poderão especificar um endereço de parâmetro nulo, em vez de especificar o endereço de uma estrutura MQBO.

#### CompCode

Tipo: MQLONG - saída

O código de conclusão; é um dos seguintes:

##### MQCC\_OK

Indica conclusão bem-sucedida.

##### MQCC\_WARNING

Aviso (conclusão parcial).

##### MQCC\_FAILED

A chamada falhou.

#### Razão

Tipo: MQLONG - saída

Se *CompCode* for MQCC\_OK:

**MQRC\_NONE**

(0, X'000') Nenhuma razão a relatar.

Se *CompCode* for MQCC\_WARNING:

**MQRC\_NO\_EXTERNAL\_PARTICIPANTES**

(2121, X'849 ') Nenhum gerenciador de recursos participante registrado.

**MQRC\_PARTICIPANT\_NOT\_AVAILABLE**

(2122, X'84A') O gerenciador de recursos participante não está disponível.

Se *CompCode* for MQCC\_FAILED:

**MQRC\_API\_EXIT\_ERROR**

(2374, X'946') Falha na saída de API.

**MQRC\_BO\_ERROR**

(2134, X'856 ') Estrutura de opções de início inválida.

**MQRC\_CALL\_IN\_PROGRESS**

(2219, X'8AB') Chamada MQI inserida antes da chamada anterior concluída.

**MQRC\_CONNECTION\_BROKEN**

(2009, X'7D9') Conexão ao gerenciador de filas perdida.

**MQRC\_ENVIRONMENT\_ERROR**

(2012, X'7DC') Chamada não válida no ambiente.

**MQRC\_HCONN\_ERROR**

(2018, X'7E2') Manipulação de conexões não válida.

**MQRC\_OPTIONS\_ERROR**

(2046, X'7FE') Opções não válidas ou não consistentes.

**MQRC\_Q\_MGR\_STOPPING**

(2162, X'872') Gerenciador de filas sendo encerrado.

**MQRC\_RESOURCE\_PROBLEM**

(2102, X'836') Recursos insuficientes do sistema disponíveis.

**MQRC\_STORAGE\_NOT\_AVAILABLE**

(2071, X'817') Armazenamento insuficiente disponível.

**MQRC\_UNEXPECTED\_ERROR**

(2195, X'893') Ocorreu um erro inesperado.

**MQRC\_UOW\_IN\_PROGRESS**

(2128, X'850 ') Unidade de trabalho já iniciada.

Para obter mais informações sobre esses códigos, consulte [Mensagens e códigos de razão](#).

## Observações de Uso

1. Use a chamada MQBEGIN para iniciar uma unidade de trabalho que é coordenada pelo gerenciador de filas e que pode envolver mudanças em recursos pertencentes a outros gerenciadores de recursos. O gerenciador de filas suporta três tipos de unidade de trabalho:
  - **Unidade de trabalho local coordenada pelo gerenciador de filas:** uma unidade de trabalho na qual o gerenciador de filas é o único gerenciador de recursos participante e, portanto, o gerenciador de filas age como o coordenador da unidade de trabalho.
    - Para iniciar esse tipo de unidade, especifique a opção MQPMO\_SYNCPOINT ou MQGMO\_SYNCPOINT na primeira chamada MQPUT, MQPUT1 ou MQGET na unidade de trabalho.
    - Para confirmar ou voltar para esse tipo de unidade de trabalho, use a chamada MQCMIT ou MQBACK
  - **Unidade de trabalho global coordenada pelo Gerenciador de Filas:** Uma unidade de trabalho na qual o gerenciador de filas age como o coordenador da unidade de trabalho, tanto para recursos do

MQ *quanto* para recursos pertencentes a outros gerenciadores de recursos. Esses gerenciadores de recursos cooperam com o gerenciador de filas para assegurar que todas as mudanças nos recursos na unidade de trabalho sejam confirmados ou restaurados juntos

- Para iniciar esse tipo de unidade de trabalho, use a chamada MQBEGIN
  - Para confirmar ou voltar esse tipo de unidade de trabalho, use as chamadas MQCMIT e MQBACK..
  - **Unidade Global Coordenada Externamente de Trabalho:** Uma unidade de trabalho na qual o gerenciador de filas é um participante, mas o gerenciador de fila não age como o coordenador da unidade de trabalho.. Em vez disso, há um coordenador da unidade de trabalho externo com o qual o gerenciador de filas coopera
    - Para iniciar este tipo de unidade de trabalho, utilize a chamada relevante fornecida pelo coordenador externo da unidade de trabalho
    - Se a chamada MQBEGIN for usada para tentar iniciar a unidade de trabalho, a chamada falhará com o código de razão MQRC\_ENVIRONMENT\_ERROR
    - Para confirmar ou voltar esse tipo de unidade de trabalho, use as chamadas de confirmação e de retorno fornecidas pelo Coordenador da unidade de trabalho externo
    - Se você usar a chamada MQCMIT ou MQBACK para confirmar ou retornar a unidade de trabalho, a chamada falhará com código de razão MQRC\_ENVIRONMENT\_ERROR.
2. Se o aplicativo terminar com mudanças não confirmados em uma unidade de trabalho, a disposição dessas mudanças dependerá se o aplicativo será encerrado normalmente ou anormalmente, ou não. Consulte as notas de uso em [“MQDISC-Desconectar gerenciador de filas”](#) na página 709 para obter detalhes adicionais
  3. Um aplicativo pode participar de apenas uma unidade de trabalho por vez.. A chamada MQBEGIN falha com o código de razão MQRC\_UOW\_IN\_PROGRESS se já houver uma unidade de trabalho existente para o aplicativo, independentemente de qual tipo de unidade de trabalho ele é.
  4. A chamada MQBEGIN não é válida em um ambiente do cliente MQI MQ . Uma tentativa de usar a chamada falha com o código de razão MQRC\_ENVIRONMENT\_ERROR
  5. Quando o gerenciador de filas está atuando como o coordenador da unidade de trabalho para unidades globais de trabalho, os gerenciadores de recursos que podem participar da unidade de trabalho são definidos no arquivo de configuração do gerenciador de filas
  6. No IBM i, os três tipos de unidade de trabalho são suportados da seguinte forma:
    - **Unidade local de trabalho coordenada pelo gerenciador de filas** pode ser usado apenas quando uma definição de confirmação não existe no nível da tarefa, ou seja o comando STRCMTCTL com o parâmetro **CMTSCOPE(\*JOB)** não deve ter sido emitido para a tarefa
    - **Unidade de trabalho global coordenada pelo gerenciador de filas** não é suportado
    - **Unidade global de trabalho coordenada externamente** pode ser usada apenas quando uma definição de confirmação existir no nível da tarefa, ou seja, o comando STRCMTCTL com o parâmetro **CMTSCOPE(\*JOB)** deve ter sido emitido para a tarefa. Se isso tiver sido feito, as operações IBM i COMMIT e ROLLBACK se aplicarão aos recursos do MQ , bem como aos recursos pertencentes a outros gerenciadores de recursos participantes

## Chamada C

```
MQBEGIN (Hconn, &BeginOptions, &CompCode, &Reason);
```

Declare os parâmetros conforme a seguir:

```
MQHCONN  Hconn;           /* Connection handle */
MQBO     BeginOptions;   /* Options that control the action of MQBEGIN */
MQLONG   CompCode;      /* Completion code */
MQLONG   Reason;        /* Reason code qualifying CompCode */
```

## Chamada COBOL

```
CALL 'MQBEGIN' USING HCONN, BEGINOPTIONS, COMPCODE, REASON.
```

Declare os parâmetros conforme a seguir:

```
** Connection handle
01 HCONN          PIC S9(9) BINARY.
** Options that control the action of MQBEGIN
01 BEGINOPTIONS.
   COPY CMQBOV.
** Completion code
01 COMPCODE       PIC S9(9) BINARY.
** Reason code qualifying COMPCODE
01 REASON         PIC S9(9) BINARY.
```

## Chamada PL/I

```
call MQBEGIN (Hconn, BeginOptions, CompCode, Reason);
```

Declare os parâmetros conforme a seguir:

```
dcl Hconn          fixed bin(31); /* Connection handle */
dcl BeginOptions  like MQBO;     /* Options that control the action of
                                MQBEGIN */
dcl CompCode      fixed bin(31); /* Completion code */
dcl Reason        fixed bin(31); /* Reason code qualifying CompCode */
```

## Chamada do Visual Basic

```
MQBEGIN Hconn, BeginOptions, CompCode, Reason
```

Declare os parâmetros conforme a seguir:

```
Dim Hconn          As Long 'Connection handle'
Dim BeginOptions  As MQBO 'Options that control the action of MQBEGIN'
Dim CompCode      As Long 'Completion code'
Dim Reason        As Long 'Reason code qualifying CompCode'
```

## MQBUFMH - Converter buffer em identificador de mensagens

A chamada de função MQBUFMH converte um buffer em uma manipulação de mensagens e é o inverso da chamada MQMHBUF.

Essa chamada usa um descritor de mensagem e propriedades MQRFH2 no buffer e os disponibiliza por meio de um identificador de mensagem.. As propriedades MQRFH2 nos dados da mensagem são, opcionalmente, removidas. Os campos *Encoding*, *CodedCharSetIde Format* do descritor de mensagens são atualizados, se necessário, para descrever corretamente o conteúdo do buffer após a remoção das propriedades.

### Sintaxe

```
MQBUFMH (Hconn, Hmsg, BufMsgHOpts, MsgDesc, BufferLength, Buffer, DataLength, Compcode, Reason)
```

### Parâmetros

#### Hconn

Tipo: MQHCONN - entrada

Esta manipulação representa a conexão ao gerenciador de filas. O valor de **Hconn** deve corresponder à manipulação de conexões que foi usada para criar a manipulação de mensagens especificada no parâmetro **Hmsg** ..

Se o identificador de mensagens tiver sido criado usando MQHC\_UNASSOCIATED\_HCONN, uma conexão válida deverá ser estabelecida no encadeamento convertendo um buffer em um identificador de mensagem. Se uma conexão válida não for estabelecida a chamada falha com MQRC\_CONNECTION\_BROKEN.

### **Hmsg**

Tipo: MQHMQSG-entrada

Este é o identificador de mensagem para o qual um buffer é necessário. O valor foi retornado por uma chamada MQCRTMH anterior.

### **BufMsgHOpts**

Tipo: MQBMHO-entrada

A estrutura MQBMHO permite que os aplicativos especifiquem opções que controlam como as manipulações de mensagens são produzidas a partir de buffers.

Consulte [“MQBMHO-Opções de manipulação de buffer para mensagem”](#) na página 280 para obter detalhes.

### **MsgDesc**

Tipo: MQMD - entrada/saída

A estrutura *MsgDesc* contém as propriedades do descritor de mensagem e descreve o conteúdo da área de buffer..

Na saída da chamada, as propriedades são removidas opcionalmente da área de buffer e, nesse caso, o descritor de mensagem é atualizado para descrever corretamente a área de buffer.

Os dados nessa estrutura devem estar no conjunto de caracteres e na codificação do aplicativo.

### **BufferLength**

Tipo: MQLONG - entrada

*BufferLength* é o comprimento da área Buffer, em bytes.

Um *BufferLength* de zero bytes é válido e indica que a área do buffer não contém dados.

### **de Saída**

Tipo: MQBYTEXBufferComprimento-entrada/saída

Estas são as opções que controlam a ação de MQBEGIN, conforme descrito em [“MQBEGIN-Iniciar unidade de trabalho”](#) na página 655.

**Buffer** define a área contendo o buffer de mensagem. Para a maioria dos dados, você deve alinhar o buffer em um limite de 4 bytes.

Se o **Buffer** contiver dados de caractere ou numéricos, configure os campos *CodedCharSetId* e *Encoding* no parâmetro **MsgDesc** para os valores apropriados para os dados; isso permite que os dados sejam convertidos, se necessário..

Se as propriedades forem localizadas no buffer de mensagem, elas serão removidas opcionalmente; posteriormente, elas se tornarão disponíveis a partir do identificador de mensagem no retorno da chamada.

Na linguagem de programação C, o parâmetro é declarado como um ponteiro para vazio, o que significa que o endereço de qualquer tipo de dados pode ser especificado como o parâmetro.

Se o parâmetro **BufferLength** for zero, **Buffer** não será referido; nesse caso, o endereço do parâmetro transmitido por programas gravados em C ou no assembler System/390 poderá ser nulo...

### **DataLength**

Tipo: MQLONG - saída

O comprimento, em bytes, do buffer que pode ter as propriedades removidas..

## CompCode

Tipo: MQLONG - saída

O código de conclusão; é um dos seguintes:

### **MQCC\_OK**

Indica conclusão bem-sucedida.

### **MQCC\_FAILED**

A chamada falhou.

## Razão

Tipo: MQLONG - saída

Se *CompCode* for MQCC\_OK:

### **MQRC\_NONE**

(0, X'000') Nenhuma razão a relatar.

Se *CompCode* for MQCC\_FAILED:

### **MQRC\_ADAPTER\_NOT\_AVAILABLE**

(2204, X'089C') Adaptador não disponível.

### **MQRC\_ADAPTER\_SERV\_LOAD\_ERROR**

(2130, X'852') Não foi possível carregar o módulo de serviço do adaptador.

### **MQRC\_ASID\_MISMATCH**

(2157, X'86D') ASIDs de início e primárias são diferentes.

### **MQRC\_BMHO\_ERROR**

(2489, X'09B9') Buffer para estrutura de opções de manipulação de mensagens não é válido.

### **MQRC\_BUFFER\_ERROR**

(2004, X'07D4') Parâmetro de buffer inválido.

### **MQRC\_BUFFER\_LENGTH\_ERROR**

(2005, X'07D5') Parâmetro de comprimento do buffer inválido.

### **MQRC\_CALL\_IN\_PROGRESS**

(2219, X'08AB') chamada MQI inserida antes da chamada anterior ser concluída.

### **MQRC\_CONNECTION\_BROKEN**

(2009, X'07D9') Conexão com o gerenciador de filas perdida.

### **MQRC\_HMSG\_ERROR**

(2460, X'099C') Identificador de mensagem inválido.

### **MQRC\_MD\_ERROR**

(2026, X'07EA') Descritor de mensagens inválido.

### **MQRC\_MSG\_HANDLE\_IN\_USE**

(2499, X'09C3') Identificador de mensagem já em uso.

### **MQRC\_OPTIONS\_ERROR**

(2046, X'07FE') Opções não válidas ou não consistentes.

### **MQRC\_RFH\_ERROR**

(2334, X'091E') MQRFH2 estrutura não válida.

### **MQRC\_RFH\_FORMAT\_ERROR**

(2421, X'0975 ') Uma pasta MQRFH2 contendo propriedades não pôde ser analisada.

### **MQRC\_UNEXPECTED\_ERROR**

(2195, X'893') Ocorreu um erro inesperado.

Para obter informações detalhadas sobre esses códigos, consulte [Mensagens e códigos de razão](#).

## Observações de Uso

As chamadas MQBUFMH não podem ser interceptadas por saídas de API-um buffer é convertido em um identificador de mensagem no espaço do aplicativo; a chamada não atinge o gerenciador de filas



## Chamada C

```
MQBUFMH (Hconn, Hmsg, &BufMsgHOpts, &MsgDesc, BufferLength, Buffer,  
&DataLength, &CompCode, &Reason);
```

Declare os parâmetros conforme a seguir:

```
MQHCONN Hconn;          /* Connection handle */  
MQHMSG Hmsg;           /* Message handle */  
MQBMHO BufMsgHOpts;   /* Options that control the action of MQBUFMH */  
MQMD MsgDesc;         /* Message descriptor */  
MQLONG BufferLength;   /* Length in bytes of the Buffer area */  
MQBYTE Buffer[n];     /* Area to contain the message buffer */  
MQLONG DataLength;    /* Length of the output buffer */  
MQLONG CompCode;     /* Completion code */  
MQLONG Reason;       /* Reason code qualifying CompCode */
```

## Chamada COBOL

```
CALL 'MQBUFMH' USING HCONN, HMSG, BUFMSGHOPTS, MSGDESC, BUFFERLENGTH,  
BUFFER, DATALENGTH, COMPCODE, REASON.
```

Declare os parâmetros conforme a seguir:

```
** Connection handle  
01 HCONN PIC S9(9) BINARY.  
** Message handle  
01 HMSG PIC S9(18) BINARY.  
** Options that control the action of MQBUFMH  
01 BUFMSGHOPTS.  
COPY CMQBMHOV.  
** Message descriptor  
01 MSGDESC.  
COPY CMQMD.  
** Length in bytes of the Buffer area  
01 BUFFERLENGTH PIC S9(9) BINARY.  
** Area to contain the message buffer  
01 BUFFER PIC X(n).  
** Length of the output buffer  
01 DATALENGTH PIC S9(9) BINARY.  
** Completion code  
01 COMPCODE PIC S9(9) BINARY.  
** Reason code qualifying COMPCODE  
01 REASON PIC S9(9) BINARY.
```

## Chamada PL/I

```
call MQBUFMH (Hconn, Hmsg, BufMsgHOpts, MsgDesc, BufferLength, Buffer,  
DataLength, CompCode, Reason);
```

Declare os parâmetros conforme a seguir:

```
dcl Hconn          fixed bin(31); /* Connection handle */  
dcl Hmsg          fixed bin(63); /* Message handle */  
dcl BufMsgHOpts   like MQBMHO;  /* Options that control the action of  
MQBUFMH */  
dcl MsgDesc       like MQMD;     /* Message descriptor */  
dcl BufferLength   fixed bin(31); /* Length in bytes of the Buffer area */  
dcl Buffer         char(n);       /* Area to contain the message buffer */  
dcl DataLength    fixed bin(31); /* Length of the output buffer */  
dcl CompCode      fixed bin(31); /* Completion code */  
dcl Reason        fixed bin(31); /* Reason code qualifying CompCode */
```

## Chamada do Assembler de Alto Nível

```
CALL MQBUFMH, (HCONN,HMSG,BUFMSGHOPTS,MSGDESC,BUFFERLENGTH,BUFFER,
              DATALENGTH,COMP CODE,REASON)
```

Declare os parâmetros conforme a seguir:

HCONN	DS	F	Connection handle
HMSG	DS	D	Message handle
BUFMSGHOPTS	CMQBMHOA	,	Options that control the action of MQBUFMH
MSGDESC	CMQMDA	,	Message descriptor
BUFFERLENGTH	DS	F	Length in bytes of the BUFFER area
BUFFER	DS	CL(n)	Area to contain the properties
DATALENGTH	DS	F	Length of the output buffer
COMP CODE	DS	F	Completion code
REASON	DS	F	Reason code qualifying COMP CODE

## MQCB-Gerenciar retorno de chamada

A chamada MQCB registra um retorno de chamada para a manipulação de objetos especificada e controla a ativação e as mudanças no retorno de chamada.

Um retorno de chamada é uma parte do código (especificado como o nome de uma função que pode ser vinculada dinamicamente ou como ponteiro de função) que é chamada pelo IBM MQ quando ocorrem determinados eventos.

Para usar MQCB e MQCTL em um cliente, deve-se estar conectado a um servidor no qual o parâmetro **SHARECNV** negociado do canal concordou com um valor diferente de zero

Os tipos de retorno de chamada que podem ser definidos são:

### Consumidor de mensagens

Uma função de retorno de chamada do consumidor de mensagens é chamada quando uma mensagem, atendendo aos critérios de seleção especificados, está disponível em uma manipulação de objetos...

Apenas uma função de retorno de chamada pode ser registrada em cada identificador de objeto. Se uma única fila tiver que ser lida com diversos critérios de seleção, a fila deverá ser aberta várias vezes e uma função do consumidor registrada em cada identificador.

### Manipulador de eventos

O manipulador de eventos é chamado para as condições que afetam todo o ambiente de retorno de chamada.

A função é chamado quando ocorre uma condição de evento, por exemplo, um gerenciador de filas ou conexão parando ou quiesce.

A função não é chamada para condições que são específicas para um único consumidor de mensagens, por exemplo MQRC\_GET\_INHIBITED; ela é chamada, no entanto, se uma função de retorno de chamada não terminar normalmente.

## Sintaxe

MQCB (*Hconn*, *Operação*, *CallbackDesc*, *Hobj*, *MsgDesc*, *GetMsgOpts*, *CompCode*, *Razão*)

## Parâmetros

### Hconn

Tipo: MQHCONN - entrada

Esta manipulação representa a conexão ao gerenciador de filas. O valor de *Hconn* foi retornado por uma chamada anterior MQCONN ou MQCONNX.

Em aplicativos z/OS for CICS , é possível especificar o seguinte valor especial para `MQHC_DEF_HCONN` para usar o identificador de conexão associado a essa unidade de execução.

### **Operação**

Tipo: MQLONG - entrada

A operação sendo processada no retorno de chamada definido para o identificador de objeto especificado. Deve-se especificar uma das opções a seguir: Para especificar mais de uma opção, inclua os valores juntos (não inclua a mesma constante mais de uma vez) ou combine os valores usando a operação OR bit a bit (se a linguagem de programação suportar operações de bit).

#### **MQOP\_REGISTER**

Defina a função de retorno de chamada para a manipulação de objetos especificada.. Esta operação define a função a ser chamada e os critérios de seleção a serem usados.

Se uma função de retorno de chamada já tiver sido definida para a manipulação de objetos, a definição será substituída. Se um erro for detectado ao substituir o retorno de chamada, a função será removida do registro..

Se um retorno de chamada for registrado na mesma função de retorno de chamada na qual ele foi removido anteriormente, isso será tratado como uma operação de substituição; quaisquer chamadas iniciais ou finais não serão chamadas.

É possível usar MQOP\_REGISTER com MQOP\_SUSPEND ou MQOP\_RESUME.

#### **MQOP\_REMOVER registro**

Pare o consumo de mensagens para a manipulação de objetos e remove a manipulação daqueles elegíveis para um retorno de chamada

Um retorno de chamada será removido automaticamente se o identificador associado for fechado.

Se MQOP\_DEREGISTER for chamado de dentro de um consumidor, e o retorno de chamada tiver uma chamada de parada definida, ele será chamado no retorno do consumidor

Se essa operação for emitida em um *Hobj* sem consumidor registrado, a chamada retornará com MQRC\_CALLBACK\_NOT\_REGISTERED.

#### **MQOP\_SUSPEND**

Suspende o consumo de mensagens para a manipulação de objetos

Se essa operação for aplicada a um manipulador de eventos, o manipulador de eventos não obterá eventos enquanto estiver suspenso e quaisquer eventos perdidos enquanto estiver no estado suspenso não serão fornecidos para a operação quando ela for retomada.

Enquanto suspensa, a função do consumidor continua a obter os retornos de chamada de tipo de controle.

#### **MQOP\_RESUME**

Retome o consumo de mensagens para a manipulação de objetos

Se essa operação for aplicada a um manipulador de eventos, o manipulador de eventos não obterá eventos enquanto estiver suspenso e quaisquer eventos perdidos enquanto estiver no estado suspenso não serão fornecidos para a operação quando ela for retomada.

### **CallbackDesc**

Tipo: MQCBD-entrada

Essa é uma estrutura que identifica a função de retorno de chamada que está sendo registrada pelo aplicativo e as opções usadas ao registrá-la,

Consulte [MQCBD](#) para obter detalhes da estrutura..

O descritor de retorno de chamada é necessário apenas para a opção MQOP\_REGISTER. Se o descritor não for necessário, o endereço do parâmetro transmitido poderá ser nulo.

### **Hobj**

Tipo: MQHOBJ - entrada

Essa manipulação representa o acesso que foi estabelecido para o objeto do qual uma mensagem deve ser consumida. Esta é uma manipulação que foi retornada de uma chamada anterior MQOPEN ou MQSUB (no parâmetro **Hobj**).

*Hobj* não é necessário ao definir uma rotina do manipulador de eventos (MQCBT\_EVENT\_HANDLER) e deve ser especificado como MQHO\_NONE.

Se *Hobj* tiver sido retornado de uma chamada MQOPEN, a fila deverá ter sido aberta com uma ou mais das seguintes opções:

- MQOO\_INPUT\_SHARED
- MQOO\_INPUT\_EXCLUSIVE
- MQOO\_INPUT\_AS\_Q\_DEF
- MQOO\_BROWSE

### **MsgDesc**

Tipo: MQMD-entrada

Essa estrutura descreve os atributos da mensagem requerida e os atributos da mensagem recuperada.

O parâmetro **MsgDesc** define os atributos das mensagens requeridas pelo consumidor e a versão do MQMD a ser transmitida para o consumidor de mensagem

O *MsgId*, *CorrelId*, *GroupId*, *MsgSeqNumber* e *Offset* no MQMD são usados para seleção de mensagens dependendo das opções especificadas no parâmetro **GetMsgOpts**.

O *Encoding* e *CodedCharSetId* serão usados para conversão de mensagem se você especificar a opção MQGMO\_CONVERT.

Consulte MQMD para obter detalhes..

*MsgDesc* é usado para MQOP\_REGISTER e se você requerer valores diferentes do padrão para quaisquer campos. O *MsgDesc* não é usado para um manipulador de eventos

Se o descritor não for necessário, o endereço do parâmetro transmitido poderá ser nulo.

Observe que, se vários consumidores forem registrados na mesma fila com seletores de sobreposição, o consumidor escolhido para cada mensagem será indefinido.

### **GetMsgOpts**

Tipo: MQGMO-entrada

O parâmetro **GetMsgOpts** controla como o consumidor de mensagens obtém mensagens.. Todas as opções desse parâmetro têm significados conforme descrito em "MQGMO-Obter opções de mensagem" na página 379, quando usado em uma chamada MQGET, exceto:

#### **MQGMO\_SET\_SIGNAL**

Esta opção não é permitida

#### **MQGMO\_BROWSE\_FIRST, MQGMO\_BROWSE\_NEXT, MQGMO\_MARK\_\***

A ordem das mensagens entregues a um consumidor de navegação é ditada pelas combinações dessas opções.. Combinações significativas são:

##### **MQGMO\_BROWSE\_FIRST**

A primeira mensagem na fila é entregue repetidamente para o consumidor Isso é útil quando o consumidor consome destrutivamente a mensagem no retorno de chamada. Use esta opção com cuidado..

##### **MQGMO\_BROWSE\_NEXT**

O consumidor recebe cada mensagem na fila, da posição atual do cursor até o final da fila ser alcançado.

##### **MQGMO\_BROWSE\_FIRST + MQGMO\_BROWSE\_NEXT**

O cursor é reconfigurado para o início da fila O consumidor recebe então cada mensagem até que o cursor atinja o final da fila.

### **MQGMO\_BROWSE\_FIRST + MQGMO\_MARK\_\***

A partir do início da fila, o consumidor recebe a primeira mensagem não marcada na fila, que é então marcada para esse consumidor. Essa combinação assegura que o consumidor possa receber novas mensagens incluídas atrás do ponto do cursor atual..

### **MQGMO\_BROWSE\_NEXT + MQGMO\_MARK\_\***

Iniciando na posição do cursor, o consumidor recebe a próxima mensagem não marcada na fila, que é então marcada para esse consumidor. Utilize esta combinação com cuidado porque as mensagens podem ser adicionadas à fila atrás da posição atual do cursor.

### **MQGMO\_BROWSE\_FIRST + MQGMO\_BROWSE\_NEXT + MQGMO\_MARK\_\***

Esta combinação não é permitida. Se usado, a chamada retornará MQRC\_OPTIONS\_ERROR

### **MQGMO\_NO\_WAIT, MQGMO\_WAIT e WaitInterval**

Essas opções controlam como o consumidor é chamado.

#### **MQGMO\_NO\_WAIT**

O consumidor nunca é chamado com MQRC\_NO\_MSG\_AVAILABLE.. O consumidor é chamado apenas para mensagens e eventos..

#### **MQGMO\_WAIT com um WaitInterval zero**

O código MQRC\_NO\_MSG\_AVAILABLE é passado para o consumidor quando não há mensagens disponíveis e o consumidor foi iniciado ou foi entregue pelo menos uma mensagem desde o último código de razão "nenhuma mensagem".

Isso impede o consumidor de pesquisar em um loop ocupado quando um intervalo de espera zero é especificado.

#### **MQGMO\_WAIT e um WaitInterval positivo..**

O consumidor é chamado após o intervalo de espera especificado com código de razão MQRC\_NO\_MSG\_AVAILABLE. Essa chamada é feita independentemente se alguma mensagem foi entregue ao consumidor.. Isso permite que o usuário execute o processamento de pulsação ou de tipo de lote

#### **MQGMO\_WAIT e WaitInterval de MQWI\_UNLIMITED**

Especifica uma espera infinita antes de retornar MQRC\_NO\_MSG\_AVAILABLE. O consumidor nunca é chamado com MQRC\_NO\_MSG\_AVAILABLE..

*GetMsgOpts* é usado apenas para MQOP\_REGISTER e se você precisar de valores diferentes do padrão para quaisquer campos O *GetMsgOpts* não é usado para um manipulador de eventos

Se o *GetMsgOpts* não for necessário, o endereço do parâmetro transmitido poderá ser nulo.. Usar esse parâmetro é o mesmo que especificar MQGMO\_DEFAULT junto com MQGMO\_FAIL\_IF QUIESCING.

Se uma manipulação de propriedades de mensagem for fornecida na estrutura MQGMO, uma cópia será fornecida na estrutura MQGMO que é transmitida para o retorno de chamada do consumidor No retorno da chamada MQCB, o aplicativo pode excluir o identificador de propriedades da mensagem

### **CompCode**

Tipo: MQLONG - saída

O código de conclusão; é um dos seguintes:

#### **MQCC\_OK**

Indica conclusão bem-sucedida.

#### **MQCC\_WARNING**

Aviso (conclusão parcial).

#### **MQCC\_FAILED**

A chamada falhou.

### **Razão**

Tipo: MQLONG - saída

Os códigos de razão na seguinte lista são aqueles que o gerenciador de filas pode retornar para o parâmetro **Reason** .

Se *CompCode* for MQCC\_OK:

**MQRC\_NONE**

(0, X'000') Nenhuma razão a relatar.

Se *CompCode* for MQCC\_FAILED:

**MQRC\_ADAPTER\_NOT\_AVAILABLE**

(2204, X'89C') Adaptador não disponível.

**MQRC\_ADAPTER\_CONV\_LOAD\_ERROR**

(2133, X'855') Impossível carregar módulos de serviços de conversão de dados.

**MQRC\_ADAPTER\_SERV\_LOAD\_ERROR**

(2130, X'852') Não foi possível carregar o módulo de serviço do adaptador.

**MQRC\_API\_EXIT\_ERROR**

(2374, X'946') Falha na saída de API.

**MQRC\_API\_EXIT\_LOAD\_ERROR**

(2183, X'887') Não foi possível carregar a saída de API.

**MQRC\_ASID\_MISMATCH**

(2157, X'86D') ASIDs de início e primárias são diferentes.

**MQRC\_BUFFER\_LENGTH\_ERROR**

(2005, X'7D5') Parâmetro de comprimento de buffer inválido.

**MQRC\_CALL\_IN\_PROGRESS**

(2219, X'8AB') Chamada MQI inserida antes da chamada anterior concluída.

**MQRC\_CALLBACK\_LINK\_ERROR**

(2487, X'9B7') Campo de tipo de retorno de chamada incorreto

**MQRC\_CALLBACK\_NOT\_REGISTERED**

(2448, X' 990 ') Não é possível cancelar registro, suspender ou continuar porque não há retorno de chamada registrado.

**MQRC\_CALLBACK\_ROUTINE\_ERROR**

(2486, X'9B6') *CallbackFunction* ou *CallbackName* deve ser especificado, mas não ambos.

**MQRC\_CALLBACK\_TYPE\_ERROR**

(2483, X'9B3') Campo de tipo de retorno de chamada incorreto

**MQRC\_CBD\_OPTIONS\_ERROR**

(2484, X'9B4') Campo de opções MQCBD incorreto.

**MQRC\_CICS\_WAIT\_FAILED**

(2140, X'85C') Pedido de espera rejeitado por CICS.

**MQRC\_CONNECTION\_BROKEN**

(2009, X'7D9') Conexão ao gerenciador de filas perdida.

**MQRC\_CONNECTION\_NOT\_AUTHORIZED**

(2217, X'8A9') Não autorizado para conexão.

**MQRC\_CONNECTION QUIESCING**

(2202, X'89A') Quiesce de conexão.

**MQRC\_CONNECTION\_STOPPING**

(2203, X'89B') Conexão sendo encerrada.

**MQRC\_CORREL\_ID\_ERROR**

(2207, X'89F') Erro de identificador de correlação.

**MQRC\_DATA\_LENGTH\_ERROR**

(2010, X'7DA') Parâmetro de comprimento de dados inválido.

**MQRC\_ENVIRONMENT\_ERROR**

(2012, X'7DC') Chamada não válida no ambiente.

**MQRC\_FUNCTION\_NOT\_SUPPORTED**

(2298, X'8FA') A função solicitada não está disponível no ambiente atual.

**MQRC\_GET\_INHIBITED**

(2016, X'7E0 ') É inibida para a fila.

**MQRC\_GLOBAL\_UOW\_CONFLICT**

(2351, X'92F') Unidades de trabalho globais conflitam.

**MQRC\_GMO\_ERROR**

(2186, X'88A') Estrutura de opções de obtenção de mensagem inválida.

**MQRC\_HANDLE\_IN\_USE\_FOR\_UOW**

(2353, X'931') Manipulação em uso para unidade de trabalho global.

**MQRC\_HCONN\_ERROR**

(2018, X'7E2') Manipulação de conexões não válida.

**MQRC\_HOBJ\_ERROR**

(2019, X'7E3') Manipulação de objetos não válida.

**MQRC\_INCONSISTENT\_BROWSE**

(2259, X'8D3') Especificação de pesquisa inconsistente.

**MQRC\_INCONSISTENT\_UOW**

(2245, X'8C5') Especificação de unidade de trabalho inconsistente.

**MQRC\_INVALID\_MSG\_UNDER\_CURSOR**

(2246, X'8C6 ') Mensagem sob o cursor inválida para recuperação.

**MQRC\_LOCAL\_UOW\_CONFLICT**

(2352, X'930') Unidade de trabalho global conflita com unidade de trabalho local.

**MQRC\_MATCH\_OPTIONS\_ERROR**

(2247, X'8C7') Opções correspondentes inválidas.

**MQRC\_MAX\_MSG\_LENGTH\_ERROR**

(2485, X'9B4') Campo *MaxMsgLength* incorreto.

**MQRC\_MD\_ERROR**

(2026, X'7EA') Descritor de mensagem inválido.

**MQRC\_MODULE\_ENTRY\_NOT\_FOUND**

(2497, X'9C1') O ponto de entrada da função especificada não pôde ser localizado no módulo.

**MQRC\_MODULE\_INVALID**

(2496, X'9C0') Módulo localizado, no entanto, é do tipo errado; não 32 bits, 64 bits ou uma biblioteca de link dinâmico válida.

**MQRC\_MODULE\_NOT\_FOUND**

(2495, X'9BF') Módulo não localizado no caminho da procura ou não autorizado a carregar.

**MQRC\_MSG\_SEQ\_NUMBER\_ERROR**

(2250, X'8CA') Número de sequência da mensagem inválido.

**MQRC\_MSG\_TOKEN\_ERROR**

(2331, X'91B') Uso de token da mensagem inválido.

**MQRC\_NO\_MSG\_AVAILABLE**

(2033, X'7F1') Nenhuma mensagem disponível.

**MQRC\_NO\_MSG\_UNDER\_CURSOR**

(2034, X'7F2') Cursor de navegação não posicionado na mensagem.

**MQRC\_NOT\_OPEN\_FOR\_BROWSE**

(2036, X'7F4') Fila não aberta para navegação.

**MQRC\_NOT\_OPEN\_FOR\_INPUT**

(2037, X'7F5') Fila não aberta para entrada.

**MQRC\_OBJECT\_CHANGED**

(2041, X'7F9') Definição de objeto alterada desde a abertura.

**MQRC\_OBJECT\_DAMAGED**

(2101, X'835') Objeto danificado.

**MQRC\_OPERATION\_ERROR**

(2206, X'89E') Código de operação incorreto na chamada API.

**MQRC\_OPTIONS\_ERROR**

(2046, X'7FE') Opções não válidas ou não consistentes.

**MQRC\_PAGESET\_ERROR**

(2193, X'891') Erro ao acessar o conjunto de dados do conjunto de páginas.

**MQRC\_Q\_DELETED**

(2052, X'804') A fila foi excluída.

**MQRC\_Q\_INDEX\_TYPE\_ERROR**

(2394, X'95A') Fila possui tipo de índice errado.

**MQRC\_Q\_MGR\_NAME\_ERROR**

(2058, X'80A') Nome do gerenciador de filas não válido ou não conhecido.

**MQRC\_Q\_MGR\_NOT\_AVAILABLE**

(2059, X'80B') Gerenciador de filas não disponível para conexão.

**MQRC\_Q\_MGR QUIESCING**

(2161, X'871') Quiesce do gerenciador de filas.

**MQRC\_Q\_MGR\_STOPPING**

(2162, X'872') Gerenciador de filas sendo encerrado.

**MQRC\_RESOURCE\_PROBLEM**

(2102, X'836') Recursos insuficientes do sistema disponíveis.

**MQRC\_SIGNAL\_OUTSTANDING**

(2069, X'815') Sinais pendentes para essa manipulação.

**MQRC\_STORAGE\_NOT\_AVAILABLE**

(2071, X'817') Armazenamento insuficiente disponível.

**MQRC\_SUPPRESSED\_BY\_EXIT**

(2109, X'83D') Chamada suprimida pelo programa de saída.

**MQRC\_SYNCPOINT\_LIMIT\_REACHED**

(2024, X'7E8') Nenhuma mensagem adicional pode ser manipulada dentro da unidade de trabalho atual.

**MQRC\_SYNCPOINT\_NOT\_AVAILABLE**

(2072, X'818') Suporte de ponto de sincronização não disponível.

**MQRC\_UNEXPECTED\_ERROR**

(2195, X'893') Ocorreu um erro inesperado.

**MQRC\_UOW\_ENLISTMENT\_ERROR**

(2354, X'932') Cadastramento na unidade de trabalho global falhou.

**MQRC\_UOW\_MIX\_NOT\_SUPPORTED**

(2355, X'933') Combinação de chamadas de unidade de trabalho não suportada.

**MQRC\_UOW\_NOT\_AVAILABLE**

(2255, X'8CF') Unidade de trabalho não disponível para o gerenciador de filas a usar.

**MQRC\_WAIT\_INTERVAL\_ERROR**

(2090, X'82A') Intervalo de Espera em MQGMO inválido.

**MQRC\_WRONG\_GMO\_VERSION**

(2256, X'8D0') Versão errada do MQGMO fornecido.

**MQRC\_WRONG\_MD\_VERSION**

(2257, X'8D1') Versão errada do MQMD fornecido.

Para obter informações detalhadas sobre esses códigos, consulte [Mensagens e códigos de razão](#).



## Observações de Uso

1. MQCB é usado para definir a ação a ser chamada para cada mensagem, correspondendo aos critérios especificados, disponível na fila. Quando a ação é processada, a mensagem é removida da fila e transmitida ao consumidor de mensagens definido ou um token de mensagem é fornecido, que é usado para recuperar a mensagem.
2. MQCB pode ser usado para definir rotinas de retorno de chamada antes de iniciar o consumo com MQCTL ou pode ser usado a partir de uma rotina de retorno de chamada.
3. Para usar MQCB de fora de uma rotina de retorno de chamada, você deve primeiro suspender o consumo de mensagens usando MQCTL e continuar o consumo posteriormente.
4. MQCB não é suportado no adaptador IMS .

## Sequência de retorno de chamada do consumidor de mensagens

É possível configurar um consumidor para chamar o retorno de chamada em pontos-chave durante o ciclo de vida do consumidor. Por exemplo:

- quando o consumidor é registrado pela primeira vez,
- quando a conexão é iniciada,
- quando a conexão for interrompida e
- quando o consumidor tem o registro removido, seja explicitamente ou implicitamente por um MQCLOSE

Verb	Significado
MQCTL (START)	Chamada MQCTL usando a Operação MQOP_START
MQCTL (STOP)	Chamada MQCTL usando a Operação MQOP_STOP
MQCTL (WAIT)	Chamada MQCTL usando a Operação MQOP_START_WAIT

Isso permite ao consumidor manter o estado associado ao consumidor. Quando um retorno de chamada é solicitado por um aplicativo, as regras para chamada do consumidor são as seguintes:

### REGISTRE-SE

É sempre o primeiro tipo de chamada do retorno de chamada

Sempre é chamado no mesmo encadeamento, como a chamada MQCB (REGISTER).

### START

É sempre chamado sincronicamente com o verbo MQCTL (START).

- Todos os retornos de chamada START são concluídos antes do retorno do verbo MQCTL (START).

Está no mesmo encadeamento que a entrega de mensagem, se THREAD\_AFFINITY for solicitado

A chamada com início não será garantida se, por exemplo, um retorno de chamada anterior emitir MQCTL (STOP) durante o MQCTL (START)

### PARAR

Nenhuma mensagem ou evento adicional será entregue após essa chamada até que a conexão seja reiniciada

Um STOP é garantido se o aplicativo foi chamado anteriormente para START ou uma mensagem ou um evento.

### DEREGISTER

É sempre o último tipo de chamada do retorno de chamada

Assegure-se de que seu aplicativo execute inicialização e limpeza baseadas em encadeamento nos retornos de chamada START e STOP. É possível executar inicialização e limpeza não baseada em encadeamento com retornos de chamadas REGISTER e DEREGISTER.

Não faça nenhuma suposição sobre a vida e a disponibilidade da linha além do que é declarado. Por exemplo, não confie em um encadeamento que permanece ativo além da última chamada para REMOVE o Registro. Da mesma forma, quando você tiver escolhido não usar THREAD\_AFFINITY, não assuma que o encadeamento existe sempre que a conexão for iniciada.

Se o seu aplicativo tiver requisitos específicos para características de encadeamento, ele sempre poderá criar um encadeamento de acordo, em seguida, usar MQCTL (WAIT). Isso tem o efeito de 'doar' o encadeamento para IBM MQ para entrega de mensagem assíncrona.

## Uso da conexão do consumidor da mensagem

É possível configurar um consumidor para chamar o retorno de chamada em pontos-chave durante o ciclo de vida do consumidor. Por exemplo:

- quando o consumidor é registrado pela primeira vez,
- quando a conexão é iniciada,
- quando a conexão for interrompida e
- quando o consumidor tem o registro removido, seja explicitamente ou implicitamente por um MQCLOSE

<i>Tabela 542. Definições de verbo MQCTL</i>	
<b>Verb</b>	<b>Significado</b>
MQCTL (START)	Chamada MQCTL usando a Operação MQOP_START
MQCTL (STOP)	Chamada MQCTL usando a Operação MQOP_STOP
MQCTL (WAIT)	Chamada MQCTL usando a Operação MQOP_START_WAIT

Isso permite ao consumidor manter o estado associado ao consumidor. Quando um retorno de chamada é solicitado por um aplicativo, as regras para chamada do consumidor são as seguintes:

### **REGISTRE-SE**

É sempre o primeiro tipo de chamada do retorno de chamada

Sempre é chamado no mesmo encadeamento, como a chamada MQCB (REGISTER).

### **START**

É sempre chamado sincronicamente com o verbo MQCTL (START).

- Todos os retornos de chamada START são concluídos antes do retorno do verbo MQCTL (START).

Está no mesmo encadeamento que a entrega de mensagem, se THREAD\_AFFINITY for solicitado

A chamada com início não será garantida se, por exemplo, um retorno de chamada anterior emitir MQCTL (STOP) durante o MQCTL (START)

### **PARAR**

Nenhuma mensagem ou evento adicional será entregue após essa chamada até que a conexão seja reiniciada

Um STOP é garantido se o aplicativo foi chamado anteriormente para START ou uma mensagem ou um evento.

### **DEREGISTER**

É sempre o último tipo de chamada do retorno de chamada

Assegure-se de que seu aplicativo execute inicialização e limpeza baseadas em encadeamento nos retornos de chamada START e STOP. É possível executar inicialização e limpeza não baseada em encadeamento com retornos de chamadas REGISTER e DEREGISTER.

Não faça nenhuma suposição sobre a vida e a disponibilidade da linha além do que é declarado. Por exemplo, não confie em um encadeamento que permanece ativo além da última chamada para REMOVE o Registro. Da mesma forma, quando você tiver escolhido não usar THREAD\_AFFINITY, não assuma que o encadeamento existe sempre que a conexão for iniciada.

Se o seu aplicativo tiver requisitos específicos para características de encadeamento, ele sempre poderá criar um encadeamento de acordo, em seguida, usar MQCTL (WAIT). Isso tem o efeito de 'doar' o encadeamento para IBM MQ para entrega de mensagem assíncrona.

## Chamada C

```
MQCB (Hconn, Operation, CallbackDesc, Hobj, MsgDesc,  
GetMsgOpts, &CompCode, &Reason);
```

Declare os parâmetros conforme a seguir:

```
MQHCONN  Hconn;          /* Connection handle */  
MQLONG   Operation;     /* Operation being processed */  
MQCBD    CallbackDesc;  /* Callback descriptor */  
MQHOBJ   HObj;          /* Object handle */  
MQMD     MsgDesc        /* Message descriptor attributes */  
MQGMO    GetMsgOpts     /* Message options */  
MQLONG   CompCode;      /* Completion code */  
MQLONG   Reason;        /* Reason code qualifying CompCode */
```

## Chamada COBOL

```
CALL 'MQCB' USING HCONN, OPERATION, CBDESC, HOBJ, MSGDESC,  
GETMSGOPTS, COMPCODE, REASON.
```

Declare os parâmetros conforme a seguir:

```
** Connection handle  
01 HCONN PIC S9(9) BINARY.  
** Operation  
01 OPERATION PIC S9(9) BINARY.  
** Callback Descriptor  
01 CBDESC.  
COPY CMQCBDV.  
01 HOBJ PIC S9(9) BINARY.  
** Message Descriptor  
01 MSGDESC.  
COPY CMQMDV.  
** Get Message Options  
01 GETMSGOPTS.  
COPY CMQGMV.  
** Completion code  
01 COMPCODE PIC S9(9) BINARY.  
** Reason code qualifying COMPCODE  
01 REASON PIC S9(9) BINARY.
```

## Chamada PL/I

```
call MQCB(Hconn, Operation, CallbackDesc, Hobj, MsgDesc, GetMsgOpts,  
CompCode, Reason)
```

Declare os parâmetros conforme a seguir:

```
dcl Hconn          fixed bin(31); /* Connection handle */  
dcl Operation     fixed bin(31); /* Operation */  
dcl CallbackDesc  like MQCBD;    /* Callback Descriptor */  
dcl Hobj          fixed bin(31); /* Object Handle */  
dcl MsgDesc       like MQMD;     /* Message Descriptor */  
dcl GetMsgOpts    like MQGMO;    /* Get Message Options */  
dcl CompCode      fixed bin(31); /* Completion code */  
dcl Reason        fixed bin(31); /* Reason code qualifying CompCode */
```

## **MQCB\_FUNCTION-Função de Retorno de Chamada**

A chamada de função MQCB\_FUNCTION é a função de retorno de chamada para manipulação de eventos e consumo de mensagens assíncronas.

A definição de chamada MQCB\_FUNCTION é fornecida exclusivamente para descrever os parâmetros que são transmitidos para a função de retorno de chamada... Nenhum ponto de entrada chamado MQCB\_FUNCTION é fornecido pelo gerenciador de filas..

A especificação da função real a ser chamada é uma entrada para a chamada [MQCB](#) e é transmitida através da estrutura [MQCBD](#) .

### **Sintaxe**

MQCB\_FUNCTION (*Hconn, MsgDesc, GetMsgOpts, Buffer, Context*)

### **Parâmetros**

#### **Hconn**

Tipo: MQHCONN - entrada

Esta manipulação representa a conexão ao gerenciador de filas. O valor de *Hconn* foi retornado por uma chamada anterior MQCONN ou MQCONNX. Em z/OS para aplicativos CICS , a chamada MQCONN pode ser omitida e o valor a seguir especificado para Hconn:

#### **MQHC\_DEF\_CONN**

Manipulação de conexões padrão.

#### **MsgDesc**

Tipo: MQMD-entrada

Esta estrutura descreve os atributos da mensagem recuperada

Consulte "[MQMD - descritor de mensagem](#)" na página 435 para obter detalhes.

A versão do MQMD transmitida é a mesma versão transmitida na chamada MQCB que definiu a função do consumidor.

O endereço do MQMD é transmitido como caracteres nulos se um MQGMO versão 4 foi usado para solicitar que um Message Handle seja retornado em vez de um MQMD.

Este é um campo de entrada para a função do consumidor de mensagens; não é relevante para uma função do manipulador de eventos.

#### **GetMsgOpts**

Tipo: MQGMO-entrada

As opções usadas para controlar as ações do consumidor de mensagem Esse parâmetro também contém informações adicionais a respeito da mensagem retornada

Consulte [MQGMO](#) para obter detalhes..

A versão de MQGMO transmitida é a versão mais recente suportada

Este é um campo de entrada para a função do consumidor de mensagens; não é relevante para uma função do manipulador de eventos.

#### **de Saída**

Tipo: MQBYTEBufferComprimento-entrada

Esta é a área que contém os dados da mensagem

Se nenhuma mensagem estiver disponível para essa chamada, ou se a mensagem não contiver dados da mensagem, o endereço do *Buffer* será transmitido como nulos

Este é um campo de entrada para a função do consumidor de mensagens; não é relevante para uma função do manipulador de eventos.

## Context

Tipo: MQCBC-entrada/saída

Essa estrutura fornece informações de contexto para as funções de retorno de chamada Consulte [“MQCBC-Contexto de retorno de chamada”](#) na página 287 para obter detalhes.

## Observações de Uso

1. Esteja ciente de que, se suas rotinas de retorno de chamada usarem serviços que poderiam atrasar ou bloquear o encadeamento, por exemplo, MQGET com espera, poderia atrasar o despacho de outros retornos de chamada
2. Uma unidade de trabalho separada não é estabelecida automaticamente para cada chamada de uma rotina de retorno de chamada, portanto, as rotinas podem emitir uma chamada de confirmação ou adiar a confirmação até que um lote lógico de trabalho seja processado. Quando o lote de trabalho é confirmado, ele confirma as mensagens para todas as funções de retorno de chamada que foram chamadas desde o último ponto de sincronização.
3. Programas chamados por CICS LINK ou CICS START recuperam parâmetros usando serviços CICS por meio de objetos denominados conhecidos como contêineres de canal. Os nomes de contêiner são os mesmos que os nomes de parâmetros. Para obter mais informações, consulte a sua documentação do CICS.
4. Rotinas de retorno de chamada podem emitir uma chamada MQDISC, mas não para sua própria conexão. Por exemplo, se uma rotina de retorno de chamada criou uma conexão, ela também poderá desconectar a conexão.
5. Uma rotina de retorno de chamada não deve, em geral, depender de ser chamada do mesmo encadeamento toda vez. Se necessário, use o MQCTLO\_THREAD\_AFFINITY quando a conexão for iniciada
6. Quando uma rotina de retorno de chamada recebe um código de razão diferente de zero, ela deve executar a ação apropriada...
7. MQCB\_FUNCTION não é suportado dentro do adaptador IMS

## MQCLOSE-Fechar objeto

A chamada MQCLOSE abandona o acesso a um objeto e é o inverso das chamadas MQOPEN e MQSUB

## Sintaxe

MQCLOSE (*Hconn*, *Hobj*, *Opções*, *CompCode*, *Motivo*)

## Parâmetros

### Hconn

Tipo: MQHCONN - entrada

Esta manipulação representa a conexão ao gerenciador de filas. O valor de *Hconn* foi retornado por uma chamada anterior MQCONN ou MQCONNX.

Em z/OS para aplicativos CICS, é possível omitir a chamada MQCONN e especificar o valor a seguir para *Hconn*::

### MQHC\_DEF\_HCONN

Manipulação de conexões padrão.

### Hobj

Tipo: MQHOBJ - entrada/saída

Esse identificador representa o objeto que está sendo fechado O objeto pode ser de qualquer tipo.. O valor de *Hobj* foi retornado por uma chamada MQOPEN anterior.

Na conclusão bem-sucedida da chamada, o gerenciador de filas configura esse parâmetro para um valor que não é um identificador válido do ambiente. Esse valor é:.

## **MQHO\_UNUSABLE\_HOBJ**

Identificador de objeto não utilizável

No z/OS, *Hobj* é configurado para um valor indefinido.

### **Opções**

Tipo: MQLONG - entrada

Este parâmetro controla como o objeto é fechado

Somente filas e assinaturas dinâmicas permanentes podem ser fechadas de mais de uma maneira, porque elas devem ser retidas ou excluídas; essas são filas com o atributo **DefinitionType** que possui o valor MQQDT\_PERMANENT\_DYNAMIC (consulte o atributo **DefinitionType** descrito em [“Atributos para filas” na página 865](#)). As opções de fechamento são resumidas neste tópico

Assinaturas duráveis podem ser mantidas ou removidas; elas são criadas usando a chamada MQSUB com a opção MQSO\_DURABLE.

Ao fechar a manipulação para um destino gerenciado (ou seja, o parâmetro **Hobj** retornado em uma chamada MQSUB que usou a opção MQSO\_MANAGED), o gerenciador de filas limpa quaisquer publicações que não foram recuperadas quando a assinatura associada também foi removida. A assinatura é removida usando a opção MQCO\_REMOVE\_SUB no parâmetro **Hsub** retornado em uma chamada do MQSUB MQCO\_REMOVE\_SUB é o comportamento padrão em MQCLOSE para uma assinatura não durável.

Ao fechar um identificador para um destino não gerenciado, você é responsável por limpar a fila para a qual as publicações são enviadas. Feche a assinatura usando MQCO\_REMOVE\_SUB primeiro e, em seguida, processe mensagens fora da fila até que nenhuma permaneça.

Você deve especificar uma opção apenas a partir do seguinte:

**Opções de fila dinâmica:** Essas opções controlam como as filas dinâmicas permanentes são fechadas.

### **MQCO\_DELETE**

A fila será excluída se uma das seguintes situações for verdadeira:

- É uma fila dinâmica permanente, criada por uma chamada MQOPEN anterior e não há mensagens na fila e nenhuma solicitação get ou put não confirmada pendente para a fila (para a tarefa atual ou qualquer outra tarefa).
- É a fila dinâmica temporária que foi criada pela chamada MQOPEN que retornou *Hobj*.. Nesse caso, todas as mensagens na fila são limpas

Em todos os outros casos, incluindo o caso em que o *Hobj* foi retornado em uma chamada MQSUB, a chamada falha com código de razão MQRC\_OPTION\_NOT\_VALID\_FOR\_TYPE e o objeto não é excluído.

No z/OS, se a fila for uma fila dinâmica que foi excluída logicamente e essa for a última manipulação para ela, a fila será excluída fisicamente. Consulte [“Observações de Uso” na página 679](#) para obter detalhes adicionais.

### **MQCO\_DELETE\_PURGE**

A fila é excluída e quaisquer mensagens nela limpas, se uma das seguintes situações for verdadeira:

- É uma fila dinâmica permanente, criada por uma chamada MQOPEN anterior e não há solicitações get ou put não confirmadas pendentes para a fila (para a tarefa atual ou qualquer outra tarefa)
- É a fila dinâmica temporária que foi criada pela chamada MQOPEN que retornou *Hobj*..

Em todos os outros casos, incluindo o caso em que o *Hobj* foi retornado em uma chamada MQSUB, a chamada falha com código de razão MQRC\_OPTION\_NOT\_VALID\_FOR\_TYPE e o objeto não é excluído.

<i>Tabela 543. Fechar opções para diferentes tipos de objeto</i>			
<b>Tipo de objeto ou fila</b>	<b>MQCO_NONE</b>	<b>MQCO_DELETE</b>	<b>MQCO_DELETE_PURGE</b>
Objeto diferente de uma fila	Retida	Inválido	Inválido
Fila predefinida	Retida	Inválido	Inválido
Fila dinâmica permanente	Retida	Excluído se vazio e nenhuma atualização pendente	Mensagens excluídas; fila excluída se nenhuma atualização pendente
Fila dinâmica temporária (chamada emitida pelo criador da fila)	Excluído	Excluído	Excluído
Fila dinâmica temporária (chamada não emitida pelo criador da fila)	Retida	Inválido	Inválido
Lista de distribuição	Retida	Inválido	Inválido
Destino de assinatura gerenciado	Retida	Inválido	Inválido
Lista de distribuição (a assinatura foi removida)	Mensagens excluídas; fila excluída	Inválido	Inválido

**Opções de encerramento de assinatura:** Essas opções controlam se as assinaturas duráveis serão removidas quando a manipulação for fechada e se as publicações ainda aguardando para serem lidas pelo aplicativo serão limpas... Essas opções são válidas somente para uso com uma manipulação de objetos retornada no parâmetro **Hsub** de uma chamada MQSUB

#### **MQCO\_KEEP\_SUB**

O identificador para a assinatura é fechado, mas a assinatura feita é mantida. As publicações continuam a ser enviadas para o destino especificado na assinatura. Essa opção será válida apenas se a assinatura tiver sido feita com a opção MQSO\_DURABLE

MQCO\_KEEP\_SUB será o padrão se a assinatura for durável

#### **MQCO\_REMOVE\_SUB**

A assinatura é removida e a manipulação para a assinatura é fechada

O parâmetro **Hobj** da chamada MQSUB não é invalidado pelo encerramento do parâmetro **Hsub** e pode continuar sendo usado para MQGET ou MQCB receber as publicações restantes. Quando o parâmetro **Hobj** da chamada MQSUB também for fechado, se ele for um destino gerenciado, quaisquer publicações não recuperadas serão removidas

MQCO\_REMOVE\_SUB será o padrão se a subscrição não for durável

Conclusão bem-sucedida de MQCO\_REMOVE\_SUB não significa que a ação foi concluída. Para verificar se essa chamada foi concluída. Consulte a etapa [DELETE SUB](#) em [Verificando se comandos assíncronos para redes distribuídas foram concluídos](#).

Essas opções de encerramento de subscrição são resumidas nas tabelas a seguir:

<i>Tabela 544. Opções para fechar um identificador de assinatura durável, mas reter a assinatura</i>	
<b>Tarefa</b>	<b>Opção de encerramento de assinatura</b>
Manter publicações em um identificador MQOPENed	MQCO_KEEP_SUB
Remover publicações em um identificador MQOPENed	Ação não permitida

Tabela 544. Opções para fechar um identificador de assinatura durável, mas reter a assinatura (continuação)

Tarefa	Opção de encerramento de assinatura
Manter publicações em um identificador MQSO_MANAGED	MQCO_KEEP_SUB
Remover publicações em um identificador MQSO_MANAGED	Ação não permitida

Para cancelar a assinatura, fechando um identificador de assinatura durável e cancelando a assinatura ou fechando um identificador de assinatura não durável, use as seguintes opções de fechamento de assinatura:

Tabela 545. Opções para cancelar assinatura

Tarefa	Opção de encerramento de assinatura
Manter publicações em um identificador MQOPENed	MQCO_REMOVE_SUB
Remover publicações em um identificador MQOPENed	Ação não permitida
Manter publicações em um identificador MQSO_MANAGED	MQCO_REMOVE_SUB

**Opções de leitura antecipada:** As opções a seguir controlam o que acontece com mensagens não persistentes que foram enviadas para o cliente antes que um aplicativo as solicitasse e ainda não foram consumidas pelo aplicativo. Essas mensagens são armazenadas no buffer de leitura antecipada do cliente aguardando para serem solicitadas pelo aplicativo e podem ser descartadas ou consumidas da fila antes que o MQCLOSE seja concluído

#### **MQCO\_IMMEDIATE**

O objeto é fechado imediatamente e quaisquer mensagens que foram enviadas ao cliente antes de um aplicativo as solicitar são descartadas e não estão disponíveis para serem consumidas por nenhum aplicativo. Esse é o valor-padrão.

#### **MQCO QUIESCE**

Uma solicitação para fechar o objeto é feita, mas se quaisquer mensagens que foram enviadas para o cliente antes que um aplicativo as solicitasse, ainda residir no buffer de leitura antecipada do cliente, a chamada MQCLOSE retornará com um aviso de MQRC\_READ\_AHEAD\_MSGS e a manipulação de objetos permanecerá válida

O aplicativo pode, então, continuar a usar o identificador de objetos para recuperar mensagens até que não mais estejam disponíveis e, em seguida, fechar o objeto novamente. Nenhuma outra mensagem é enviada para o cliente antes de um aplicativo solicitar, a leitura antecipada está agora desativada.

Os aplicativos são aconselhados a usar MQCO QUIESCE em vez de tentar atingir um ponto em que não há mais mensagens no buffer de leitura antecipada do cliente, porque uma mensagem poderia chegar entre a última chamada MQGET e o MQCLOSE a seguir, que seria descartada se MQCO\_IMMEDIATE fosse usado

Se um MQCLOSE com MQCO QUIESCE for emitido de dentro de uma função de retorno de chamada assíncrona, o mesmo comportamento de leitura de mensagens de avanço se aplicará. Se o aviso MQRC\_READ\_AHEAD\_MSGS for retornado, a função de retorno de chamada será chamada pelo menos mais uma vez. Quando a última mensagem restante que foi lida antecipadamente tiver sido transmitida para a função de retorno de chamada, o campo MQCBC ConsumerFlags será configurado como MQCBCF\_READA\_BUFFER\_EMPTY.

**Opção padrão:** Se você não precisar de nenhuma das opções descritas anteriormente, poderá usar a opção a seguir:

#### **MQCO\_NONE**

Nenhum processamento de fechamento opcional necessário.



Isso deve ser especificado para:

- Objetos diferentes de filas
- Filas Predefinidas
- Filas dinâmicas temporárias (mas somente nos casos em que *Hobj* não é o identificador retornado pela chamada MQOPEN que criou a fila).
- Listas de distribuição

Em todos os casos acima, o objeto é mantido e não excluído.

Se esta opção for especificada para uma fila dinâmica temporária:

- A fila será excluída, se tiver sido criada pela chamada MQOPEN que retornou *Hobj* ; todas as mensagens que estão na fila são limpas
- Em todos os outros casos, a fila (e quaisquer mensagens nela) são retidos..

Se essa opção for especificada para uma fila dinâmica permanente a fila será retida e não excluída.

No z/OS, se a fila for uma fila dinâmica que foi excluída logicamente e essa for a última manipulação para ela, a fila será excluída fisicamente Consulte [“Observações de Uso” na página 679](#) para obter detalhes adicionais.

### CompCode

Tipo: MQLONG - saída

O código de conclusão; é um dos seguintes:

#### **MQCC\_OK**

Indica conclusão bem-sucedida.

#### **MQCC\_WARNING**

Aviso (conclusão parcial).

#### **MQCC\_FAILED**

A chamada falhou.

### Razão

Tipo: MQLONG - saída

Os códigos de razão listados são aqueles que o Gerenciador de Filas pode retornar para o parâmetro

#### **Reason**

Se *CompCode* for MQCC\_OK:

#### **MQRC\_NONE**

(0, X'000') Nenhuma razão a relatar.

Se *CompCode* for MQCC\_WARNING:

#### **MQRC\_INCOMPLETE\_GROUP**

(2241, X'8C1') Grupo de mensagens não concluído.

#### **MQRC\_INCOMPLETE\_MSG**

(2242, X'8C2') Mensagem lógica não concluída.

#### **MQRC\_READ\_AHEAD\_MSGS**

(nnnn, X'xxx ') O cliente leu mensagens que ainda não foram consumidas pelo aplicativo.

Se *CompCode* for MQCC\_FAILED:

#### **MQRC\_ADAPTER\_NOT\_AVAILABLE**

(2204, X'89C') Adaptador não disponível.

#### **MQRC\_ADAPTER\_SERV\_LOAD\_ERROR**

(2130, X'852') Não foi possível carregar o módulo de serviço do adaptador.

#### **MQRC\_API\_EXIT\_ERROR**

(2374, X'946') Falha na saída de API.

**MQRC\_API\_EXIT\_LOAD\_ERROR**

(2183, X'887') Não foi possível carregar a saída de API.

**MQRC\_ASID\_MISMATCH**

(2157, X'86D') ASIDs de início e primárias são diferentes.

**MQRC\_CALL\_IN\_PROGRESS**

(2219, X'8AB') Chamada MQI inserida antes da chamada anterior concluída.

**MQRC\_CF\_NOT\_AVAILABLE**

(2345, X' 929 ') Recurso de acoplamento não disponível.

**MQRC\_CF\_STRUC\_FAILED**

(2373, X'945') Falha na estrutura do recurso de acoplamento.

**MQRC\_CF\_STRUC\_IN\_USE**

(2346, X'92A') Estrutura de recurso de acoplamento em uso.

**MQRC\_CICS\_WAIT\_FAILED**

(2140, X'85C') Pedido de espera rejeitado por CICS.

**MQRC\_CONNECTION\_BROKEN**

(2009, X'7D9') Conexão ao gerenciador de filas perdida.

**MQRC\_CONNECTION\_NOT\_AUTHORIZED**

(2217, X'8A9') Não autorizado para conexão.

**MQRC\_CONNECTION\_STOPPING**

(2203, X'89B') Conexão sendo encerrada.

**MQRC\_DB2\_NOT\_AVAILABLE**

(2342, X' 926 ') Db2 subsistema não disponível.

**MQRC\_HCONN\_ERROR**

(2018, X'7E2') Manipulação de conexões não válida.

**MQRC\_HOBJ\_ERROR**

(2019, X'7E3') Manipulação de objetos não válida.

**MQRC\_NOT\_AUTHORIZED**

(2035, X'7F3') Não autorizado para acesso.

**MQRC\_OBJECT\_DAMAGED**

(2101, X'835') Objeto danificado.

**MQRC\_OPTION\_NOT\_VALID\_FOR\_TYPE**

(2045, X'7FD') Em uma chamada MQOPEN ou MQCLOSE: opção não válida para o tipo de objeto.

**MQRC\_OPTIONS\_ERROR**

(2046, X'7FE') Opções não válidas ou não consistentes.

**MQRC\_PAGESET\_ERROR**

(2193, X'891') Erro ao acessar o conjunto de dados do conjunto de páginas.

**MQRC\_Q\_MGR\_NAME\_ERROR**

(2058, X'80A') Nome do gerenciador de filas não válido ou não conhecido.

**MQRC\_Q\_MGR\_NOT\_AVAILABLE**

(2059, X'80B') Gerenciador de filas não disponível para conexão.

**MQRC\_Q\_MGR\_STOPPING**

(2162, X'872') Gerenciador de filas sendo encerrado.

**MQRC\_Q\_NOT\_EMPTY**

(2055, X'807 ') A fila contém uma ou mais mensagens ou solicitações put ou get não confirmados.

**MQRC\_RESOURCE\_PROBLEM**

(2102, X'836') Recursos insuficientes do sistema disponíveis.

**MQRC\_SECURITY\_ERROR**

(2063, X'80F') Ocorreu um erro na segurança.

**MQRC\_STORAGE\_NOT\_AVAILABLE**

(2071, X'817') Armazenamento insuficiente disponível.

### **MQRC\_SUPPRESSED\_BY\_EXIT**

(2109, X'83D') Chamada suprimida pelo programa de saída.

### **MQRC\_UNEXPECTED\_ERROR**

(2195, X'893') Ocorreu um erro inesperado.

Para obter informações detalhadas sobre esses códigos, consulte [Mensagens e códigos de razão](#).

## **Observações de Uso**

1. Quando um aplicativo emite a chamada MQDISC, ou termina normalmente ou de forma anormal, quaisquer objetos que foram abertos pelo aplicativo e ainda estão abertos são fechados automaticamente com a opção MQCO\_NONE
2. Os pontos a seguir se aplicarão se o objeto que estiver sendo fechado for uma *fila*:
  - Se as operações na fila forem executadas como parte de uma unidade de trabalho, a fila poderá ser encerrada antes ou depois que o ponto de sincronização ocorrer sem afetar o resultado do ponto de sincronização. Se a fila for acionada, executar um retrocesso antes de fechar a fila pode fazer com que uma mensagem do acionador seja emitida. Para obter mais informações sobre mensagens do acionador, consulte [Propriedades de mensagens do acionador](#).
  - Se a fila foi aberta com a opção MQOO\_BROWSE, o cursor de procura será destruído.. Se a fila for reaberta com a opção MQOO\_BROWSE, um novo cursor de procura será criado (consulte [MQOO\\_BROWSE](#) ).
  - Se uma mensagem estiver atualmente bloqueada para esta manipulação no momento da chamada MQCLOSE, o bloqueio será liberado (consulte [MQGMO\\_LOCK](#) ).
  - No z/OS, se houver uma solicitação MQGET com a opção MQGMO\_SET\_SIGNAL pendente com relação à manipulação de fila que está sendo fechada, a solicitação será cancelada (consulte [MQGMO\\_SET\\_SIGNAL](#) ). As solicitações de sinal para a mesma fila, mas alojadas em diferentes identificadores (*Hobj*) não são afetadas (a menos que uma fila dinâmica esteja sendo excluída, nesse caso, elas também serão canceladas).
3. Os seguintes pontos se aplicam se o objeto que está sendo fechado for uma *fila dinâmica* (permanente ou temporária):
  - Para uma fila dinâmica, é possível especificar as opções MQCO\_DELETE e MQCO\_DELETE\_PURGE independentemente das opções especificadas na chamada MQOPEN correspondente.
  - Quando uma fila dinâmica é excluída todas as chamadas MQGET com a opção MQGMO\_WAIT que estão pendentes na fila são canceladas e o código de razão MQRC\_Q\_DELETED é retornado. Consulte [MQGMO\\_WAIT](#).

Embora os aplicativos não possam acessar uma fila excluída, a fila não é removida do sistema e os recursos associados não são liberados até que todos os identificadores que fazem referência à fila tenham sido fechados e todas as unidades de trabalho que afetam a fila tenham sido confirmadas ou restauradas.

No z/OS, uma fila que foi logicamente excluída, mas ainda não removida do sistema impede a criação de uma nova fila com o mesmo nome que a fila excluída; a chamada MQOPEN falha com o código de razão MQRC\_NAME\_IN\_USE neste caso. Além disso, essa fila ainda pode ser exibida usando comandos MQSC, embora não possa ser acessada por aplicativos.

- Quando uma fila dinâmica permanente é excluída, se o identificador *Hobj* especificado na chamada MQCLOSE não for aquele que foi retornado pela chamada MQOPEN que criou a fila, será feita uma verificação de que o identificador de usuário que foi usado para validar a chamada MQOPEN está autorizado a excluir a fila.. Se a opção MQOO\_ALTERNATE\_USER\_AUTHORITY foi especificada na chamada MQOPEN, o identificador de usuário verificado é o *AlternateUserId*...

Essa verificação não será executada se:

- O identificador especificado é aquele retornado pela chamada MQOPEN que criou a fila..
- A fila sendo excluída é uma fila dinâmica temporária.

- Quando uma fila dinâmica temporária é fechada, se o identificador *Hobj* especificado na chamada MQCLOSE for aquele que foi retornado pela chamada MQOPEN que criou a fila, a fila será excluída. Isso ocorre independentemente das opções de fechamento especificadas na chamada MQCLOSE.. Se houver mensagens na fila, elas serão descartadas; nenhuma mensagem de relatório será gerada.

Se houver unidades de trabalho não confirmadas que afetam a fila, a fila e suas mensagens ainda são excluídas, mas as unidades de trabalho não falham. No entanto, conforme descrito anteriormente, os recursos associados às unidades de trabalho não são liberados até que cada uma delas tenha sido confirmada ou restaurada.

4. Os seguintes pontos se aplicam se o objeto que está sendo fechado for uma *lista de distribuição*:

- A única opção de fechamento válida para uma lista de distribuição é MQCO\_NONE; a chamada falha com o código de razão MQRC\_OPTIONS\_ERROR ou MQRC\_OPTION\_NOT\_VALID\_FOR\_TYPE se quaisquer outras opções forem especificadas.
- Quando uma lista de distribuição é fechada, os códigos de conclusão individuais e os códigos de razão não são retornados para as filas na lista; apenas os parâmetros **CompCode** e **Reason** da chamada estão disponíveis para propósitos de diagnóstico.

Se ocorrer uma falha ao fechar uma das filas, o gerenciador de filas continua o processamento e tenta fechar as filas restantes na lista de distribuição. Os parâmetros **CompCode** e **Reason** da chamada são configurados para retornar informações que descrevem a falha. É possível que o código de conclusão seja MQCC\_FAILED, mesmo que a maioria das filas tenha sido fechada com êxito. A fila que encontrou o erro não é identificada.

Se houver uma falha em mais de uma fila, não será definido qual falha será relatada nos parâmetros **CompCode** e **Reason**.

## Chamada C

```
MQCLOSE (Hconn, &Hobj, Options, &CompCode, &Reason);
```

Declare os parâmetros conforme a seguir:

```
MQHCONN  Hconn;      /* Connection handle */
MQHOBJ   Hobj;       /* Object handle */
MQLONG   Options;    /* Options that control the action of MQCLOSE */
MQLONG   CompCode;   /* Completion code */
MQLONG   Reason;     /* Reason code qualifying CompCode */
```

## Chamada COBOL

```
CALL 'MQCLOSE' USING HCONN, HOBJ, OPTIONS, COMPCODE, REASON.
```

Declare os parâmetros conforme a seguir:

```
** Connection handle
01 HCONN      PIC S9(9) BINARY.
** Object handle
01 HOBJ       PIC S9(9) BINARY.
** Options that control the action of MQCLOSE
01 OPTIONS    PIC S9(9) BINARY.
** Completion code
01 COMPCODE   PIC S9(9) BINARY.
** Reason code qualifying COMPCODE
01 REASON     PIC S9(9) BINARY.
```

## Chamada PL/I

```
call MQCLOSE (Hconn, Hobj, Options, CompCode, Reason);
```

Declare os parâmetros conforme a seguir:

```
dcl Hconn      fixed bin(31); /* Connection handle */
dcl Hobj       fixed bin(31); /* Object handle */
dcl Options    fixed bin(31); /* Options that control the action of
                               MQCLOSE */
dcl CompCode   fixed bin(31); /* Completion code */
dcl Reason     fixed bin(31); /* Reason code qualifying CompCode */
```

## Chamada do Assembler de Alto Nível

```
CALL MQCLOSE, (HCONN, HOBJ, OPTIONS, COMPCODE, REASON)
```

Declare os parâmetros conforme a seguir:

```
HCONN      DS F Connection handle
HOBJ       DS F Object handle
OPTIONS    DS F Options that control the action of MQCLOSE
COMPCODE   DS F Completion code
REASON     DS F Reason code qualifying COMPCODE
```

## Chamada do Visual Basic

```
MQCLOSE Hconn, Hobj, Options, CompCode, Reason
```

Declare os parâmetros conforme a seguir:

```
Dim Hconn As Long 'Connection handle'
Dim Hobj As Long 'Object handle'
Dim Options As Long 'Options that control the action of MQCLOSE'
Dim CompCode As Long 'Completion code'
Dim Reason As Long 'Reason code qualifying CompCode'
```

## MQCMIT-Confirmar mudanças

A chamada MQCMIT indica ao gerenciador de filas que o aplicativo atingiu um ponto de sincronização e que todas as obtenções e colocações de mensagens que ocorreram desde o último ponto de sincronização devem ser permanentes.

Mensagens colocadas como parte de uma unidade de trabalho são disponibilizadas para outros aplicativos; mensagens recuperadas como parte de uma unidade de trabalho são excluídas.

- ▶ **z/OS** No z/OS, a chamada é usada apenas por programas em lote (incluindo IMS programas DL/I em lote).

## Sintaxe

MQCMIT (*Hconn*, *CompCode*, *Reason*)

## Parâmetros

### Hconn

Tipo: MQHCONN - entrada

Esta manipulação representa a conexão ao gerenciador de filas. O valor de *Hconn* foi retornado por uma chamada anterior MQCONN ou MQCONNX.

### **CompCode**

Tipo: MQLONG - saída

O código de conclusão; é um dos seguintes:

#### **MQCC\_OK**

Indica conclusão bem-sucedida.

#### **MQCC\_WARNING**

Aviso (conclusão parcial).

#### **MQCC\_FAILED**

A chamada falhou.

### **Razão**

Tipo: MQLONG - saída

Os códigos de razão listados são aqueles que o Gerenciador de Filas pode retornar para o parâmetro **Reason**

Se *CompCode* for MQCC\_OK:

#### **MQRC\_NONE**

(0, X'000') Nenhuma razão a relatar.

Se *CompCode* for MQCC\_WARNING:

#### **MQRC\_BACKED\_OUT**

(2003, X'7D3') Unidade de trabalho retornada.

#### **MQRC\_OUTCOME\_PENDING**

(2124, X'84C') Resultado da operação de confirmação pendente.

Se *CompCode* for MQCC\_FAILED:

#### **MQRC\_ADAPTER\_SERV\_LOAD\_ERROR**

(2130, X'852') Não foi possível carregar o módulo de serviço do adaptador.

#### **MQRC\_API\_EXIT\_ERROR**

(2374, X'946') Falha na saída de API.

#### **MQRC\_ASID\_MISMATCH**

(2157, X'86D') ASIDs de início e primárias são diferentes.

#### **MQRC\_CALL\_IN\_PROGRESS**

(2219, X'8AB') Chamada MQI inserida antes da chamada anterior concluída.

#### **MQRC\_CALL\_INTERRUPTED**

(2549, X'9F5') MQPUT ou MQCMIT foi interrompido e o processamento de reconexão não pode restabelecer um resultado definitivo.

#### **MQRC\_CF\_STRUC\_IN\_USE**

(2346, X'92A') Estrutura de recurso de acoplamento em uso.

#### **MQRC\_CONNECTION\_BROKEN**

(2009, X'7D9') Conexão ao gerenciador de filas perdida.

#### **MQRC\_ENVIRONMENT\_ERROR**

(2012, X'7DC') Chamada não válida no ambiente.

#### **MQRC\_HCONN\_ERROR**

(2018, X'7E2') Manipulação de conexões não válida.

#### **MQRC\_OBJECT\_DAMAGED**

(2101, X'835') Objeto danificado.

#### **MQRC\_OUTCOME\_MIXED**

(2123, X'84B') O resultado da operação de confirmação ou de retorno é combinado.

**MQRC\_Q\_MGR\_STOPPING**

(2162, X'872') Gerenciador de filas sendo encerrado.

**MQRC\_RECONNECT\_FAILED**

(2548, X'9F4') Após a reconexão, ocorreu um erro ao restabelecer os identificadores para uma conexão reconectável...

**MQRC\_RESOURCE\_PROBLEM**

(2102, X'836') Recursos insuficientes do sistema disponíveis.

**MQRC\_STORAGE\_MEDIUM\_FULL**

(2192, X'890') O meio de armazenamento externo está cheio.

**MQRC\_STORAGE\_NOT\_AVAILABLE**

(2071, X'817') Armazenamento insuficiente disponível.

**MQRC\_UNEXPECTED\_ERROR**

(2195, X'893') Ocorreu um erro inesperado.

Para obter informações detalhadas sobre esses códigos, consulte [Mensagens e códigos de razão](#).

## Observações de Uso

1. Use essa chamada apenas quando o próprio gerenciador de filas coordenar a unidade de trabalho. Ele pode ser:
  - Uma unidade de trabalho local, em que as mudanças afetam apenas os recursos do IBM MQ
  - Uma unidade de trabalho global, na qual as mudanças podem afetar recursos pertencentes a outros gerenciadores de recursos, bem como afetar recursos do IBM MQ .

Para obter detalhes adicionais sobre as unidades de trabalho locais e globais, consulte [“MQBEGIN- Iniciar unidade de trabalho”](#) na página 655


2. Em ambientes em que o gerenciador de filas não coordena a unidade de trabalho, a chamada de confirmação apropriada deve ser usada em vez de MQCMIT. O ambiente também pode suportar uma confirmação implícita causada pelo aplicativo sendo finalizado normalmente.
  - No z/OS, use as seguintes chamadas:
    - Programas em lote (incluindo IMS programas DL/I em lote) podem usar a chamada MQCMIT se a unidade de trabalho afetar apenas recursos IBM MQ . No entanto, se a unidade de trabalho afetar os recursos IBM MQ e os recursos pertencentes a outros gerenciadores de recursos (por exemplo, Db2 ), use a chamada SRRCMIT fornecida pelo z/OS Recoverable Resource Service (RRS). A chamada SRRCMIT confirma as mudanças nos recursos pertencentes aos gerenciadores de recursos que foram ativados para coordenação RRS.
    - Os aplicativos CICS devem usar o comando EXEC CICS SYNCPOINT para confirmar a unidade de trabalho explicitamente. Como alternativa, terminar a transação resulta em uma confirmação implícita da unidade de trabalho. A chamada MQCMIT não pode ser usada para aplicativos CICS ..
    - Os aplicativos IMS (diferentes de programas DL/I em lote) devem usar chamadas IMS como GU e CHKP para confirmar a unidade de trabalho. A chamada MQCMIT não pode ser usada para aplicativos IMS (diferentes de programas DL/I em lote).
  - No IBM i, use essa chamada para unidades locais de trabalho coordenadas pelo gerenciador de fila.. Isso significa que uma definição de confirmação não deve existir no nível da tarefa, ou seja, o comando STRCMTCTL com o parâmetro **CMTSCOPE(\*JOB)** não deve ter sido emitido para a tarefa
3. Se um aplicativo terminar com mudanças não confirmados em uma unidade de trabalho, a disposição dessas mudanças dependerá se o aplicativo será encerrado normalmente ou de forma anormal. Consulte [Notas de uso do MQDISC](#) para obter detalhes adicionais
4. Quando um aplicativo coloca ou obtém mensagens em grupos ou segmentos de mensagens lógicas, o gerenciador de filas retém informações relacionadas ao grupo de mensagens e mensagens lógicas para as últimas chamadas MQPUT e MQGET bem-sucedidas. Essas informações estão associadas ao identificador de filas e incluem itens como:
  - Os valores dos campos *GroupId*, *MsgSeqNumber*, *Offset* e *MsgFlags* em MQMD..

- Indica se a mensagem faz parte de uma unidade de trabalho
- Para a chamada MQPUT: se a mensagem é persistente ou não persistente.

Quando uma unidade de trabalho é confirmada, o gerenciador de filas retém as informações do grupo e do segmento e o aplicativo pode continuar colocando ou obtendo mensagens no grupo de mensagens ou na mensagem lógica atual

Reter as informações do grupo e do segmento quando uma unidade de trabalho é confirmada permite que o aplicativo difunda um grupo de mensagens grandes ou uma mensagem lógica grande que consiste em muitos segmentos em várias unidades de trabalho... Usar várias unidades de trabalho será vantajoso se o gerenciador de filas locais tiver apenas armazenamento de fila limitado. No entanto, o aplicativo deve manter informações suficientes para reiniciar colocando ou obtendo mensagens no ponto correto se ocorrer uma falha do sistema. Para obter detalhes sobre como reiniciar no ponto correto após uma falha do sistema, consulte [MQPMO\\_LOGICAL\\_ORDER](#) e [MQGMO\\_LOGICAL\\_ORDER](#).

As notas de uso restantes se aplicam apenas quando o gerenciador de filas coordena as unidades de trabalho:

5. Uma unidade de trabalho tem o mesmo escopo que uma manipulação de conexões; todas as chamadas IBM MQ que afetam uma determinada unidade de trabalho devem ser executadas usando a mesma manipulação de conexões Chamadas emitidas usando um identificador de conexão diferente (por exemplo, chamadas emitidas por outro aplicativo) afetam uma unidade de trabalho diferente. Consulte o parâmetro **Hconn** descrito em MQCONN para obter informações sobre o escopo de identificadores de conexão
6. Apenas as mensagens que foram colocadas ou recuperadas como parte da unidade de trabalho atual são afetada por esta chamada
7. Um aplicativo de longa execução que emite chamadas MQGET ou MQPUT ou MQPUT1 dentro de uma unidade de trabalho, mas que nunca emite uma chamada de confirmação ou de retorno, pode preencher filas com mensagens que não estão disponíveis para outros aplicativos. Para se proteger contra isso, o administrador deve configurar o atributo do gerenciador de filas **MaxUncommittedMsgs** para um valor que seja baixo o suficiente para evitar que aplicativos runaway preencham as filas, mas alto o suficiente para permitir que os aplicativos de sistema de mensagens esperados funcionem corretamente.
8.  Em sistemas AIX, Linux, and Windows , se o parâmetro **Reason** for MQRC\_CONNECTION\_BROKEN (com um *CompCode* de MQCC\_FAILED) ou MQRC\_UNEXPECTED\_ERROR, é possível que a unidade de trabalho tenha sido confirmada com sucesso.

## Chamada C

```
MQCMIT (Hconn, &CompCode, &Reason);
```

Declare os parâmetros conforme a seguir:

```
MQHCONN  Hconn;      /* Connection handle */
MQLONG   CompCode;  /* Completion code */
MQLONG   Reason;    /* Reason code qualifying CompCode */
```

## Chamada COBOL

```
CALL 'MQCMIT' USING HCONN, COMPCODE, REASON.
```

Declare os parâmetros conforme a seguir:

```
** Connection handle
```



```

01 HCONN      PIC S9(9) BINARY.
** Completion code
01 COMPCODE  PIC S9(9) BINARY.
** Reason code qualifying COMPCODE
01 REASON    PIC S9(9) BINARY.

```

## Chamada PL/I

```
call MQCMIT (Hconn, CompCode, Reason);
```

Declare os parâmetros conforme a seguir:

```

dcl Hconn      fixed bin(31); /* Connection handle */
dcl CompCode   fixed bin(31); /* Completion code */
dcl Reason     fixed bin(31); /* Reason code qualifying CompCode */

```

## Chamada do Assembler de Alto Nível

```
CALL MQCMIT,(HCONN,COMPCODE,REASON)
```

Declare os parâmetros conforme a seguir:

```

HCONN      DS F Connection handle
COMPCODE   DS F Completion code
REASON     DS F Reason code qualifying COMPCODE

```

## Chamada do Visual Basic

```
MQCMIT Hconn, CompCode, Reason
```

Declare os parâmetros conforme a seguir:

```

Dim Hconn      As Long 'Connection handle'
Dim CompCode   As Long 'Completion code'
Dim Reason     As Long 'Reason code qualifying CompCode'

```

## MQCONN - Conectar o gerenciador de filas

A chamada MQCONN conecta um programa de aplicativo a um gerenciador de filas.

Ele fornece uma manipulação de conexões do gerenciador de filas, que o aplicativo usa em chamadas de enfileiramento de mensagens subsequentes.

- No z/OS, os aplicativos CICS não precisam emitir essa chamada. Esses aplicativos são conectados automaticamente ao gerenciador de filas ao qual o sistema CICS está conectado. No entanto, as chamadas MQCONN e MQDISC ainda são aceitas de aplicativos CICS.
- No IBM i, os aplicativos devem usar a chamada MQCONN ou MQCONNX para se conectar ao gerenciador de filas e a chamada MQDISC para se desconectar do gerenciador de filas.

Uma conexão do cliente não pode ser estabelecida em uma instalação de servidor apenas e uma conexão local não pode ser estabelecida em uma instalação de cliente apenas.

## Sintaxe

```
MQCONN (QMgrName, Hconn, CompCode, Reason)
```

## Parâmetros

### QMgrName

Tipo: MQCHAR48 - entrada

Este é o nome do gerenciador de filas ao qual o aplicativo deseja se conectar. O nome pode conter os seguintes caracteres:

- Caracteres alfabéticos maiúsculos (A a Z)
- Caracteres alfabéticos minúsculos (a a z)
- Dígitos numéricos (0 a 9)
- Ponto (.), barra (/), sublinhado (\_), porcentagem (%)

O nome não deve conter espaços em branco iniciais ou integrados, mas pode conter rastreios de espaço em branco. Um caractere nulo pode ser usado para indicar o término de dados significativos no nome; o nulo e quaisquer caracteres subsequentes são tratados como espaços em branco. As restrições a seguir aplicam-se aos ambientes indicados:

- Em sistemas que usam EBCDIC Katakana, caracteres minúsculos não podem ser usados.
- No z/OS, nomes que iniciam ou terminam com um sublinhado não podem ser processados pelas operações e pelos painéis de controle. Por esse motivo, evite esses nomes.
- No IBM i, coloque os nomes contendo caracteres minúsculos, barra ou percentual entre aspas quando especificados em comandos. Não especifique essas aspas no parâmetro **QMgrName**

Se o nome consistir inteiramente de espaços em branco, o nome do gerenciador de filas *padrão* será usado. No entanto, observe o uso de nomes de gerenciadores de filas em branco descritos na seção sobre aplicativos do IBM MQ MQI client

O nome especificado para *QMgrName* deve ser o nome de um gerenciador de filas *conectável* ou, se os grupos de gerenciadores de fila estiverem sendo usados, o nome do grupo de gerenciadores de filas ..

No z/OS, os gerenciadores de filas ao quais é possível se conectar são determinados pelo ambiente:

- Para o CICS, é possível usar somente o gerenciador de filas ao qual o sistema CICS está conectado. O parâmetro **QMgrName** ainda deve ser especificado, mas seu valor é ignorado; caracteres em branco são uma opção adequada.
- Para o IMS, somente os gerenciadores de filas listados na tabela de definição do subsistema (CSQQDEFV) e na tabela SSM no IMS são conectáveis (veja a observação de uso [6](#)).
- Para o lote e a TSO (Time Sharing Option) do z/OS, somente os gerenciadores de filas residentes no mesmo sistema que o aplicativo são conectáveis (veja a observação de uso [6](#)).

**Grupos de filas compartilhadas:** Em sistemas em que existem vários gerenciadores de filas e são configurados para formar um grupo de filas compartilhadas, o nome do grupo de filas compartilhadas pode ser especificado para *QMgrName* no lugar do nome de um gerenciador de filas. Isso permite que o aplicativo se conecte a *qualquer* gerenciador de filas disponível no grupo de filas compartilhadas e que esteja na mesma imagem do z/OS que o aplicativo. O sistema também pode ser configurado para que o uso de um *QMgrName* em branco se conecte ao grupo de filas compartilhadas em vez do gerenciador de filas padrão.

Se *QMgrName* especificar o nome do grupo de filas compartilhadas, mas também houver um gerenciador de fila com esse nome no sistema, a conexão será feita com o último em preferência ao primeiro... Somente se essa conexão falhar, será feita uma tentativa de conexão com um dos gerenciadores de filas no grupo de filas compartilhadas.

Se a conexão for bem-sucedida, será possível usar a manipulação retornada pela chamada MQCONN ou MQCONNX para acessar *todos* os recursos (compartilhados e não compartilhados) pertencentes ao gerenciador de filas com o qual a conexão foi feita. O acesso a esses recursos está sujeito aos controles típicos de autorização.

Se o aplicativo emitir duas chamadas MQCONN ou MQCONNX para estabelecer conexões simultâneas e uma ou ambas as chamadas especificarem o nome do grupo de filas compartilhadas, a

segunda chamada retornará o código de conclusão MQCC\_WARNING e o código de razão MQRC\_ALREADY\_CONNECTED quando ele se conectar ao mesmo gerenciador de filas da primeira chamada.

Grupos de filas compartilhadas são suportados somente no z/OS. A conexão com um grupo de filas compartilhadas é suportada apenas nos ambientes em lote, em lote RRS, CICS e TSO. Para o CICS, é possível usar somente o grupo de filas compartilhadas ao qual o sistema CICS está conectado. Deve-se ainda especificar o parâmetro **QMGRName**, mas seu valor é ignorado; caracteres em branco são uma opção adequada.



**Atenção:** O IMS não pode se conectar a um grupo de filas compartilhadas.

**IBM MQ MQI client aplicativos:** para aplicativos IBM MQ MQI client, uma conexão é tentada para cada definição de canal de conexão do cliente com o nome do gerenciador de filas especificado, até que uma seja bem-sucedida... O gerenciador de filas, no entanto, deve ter o mesmo nome que o nome especificado. Se um nome em branco for especificado, cada canal de conexão do cliente com um nome do gerenciador de filas em branco será tentado até que um seja bem-sucedido; neste caso, não há verificação com relação ao nome real do gerenciador de filas.

Os aplicativos clientes IBM MQ não são suportados no z/OS, mas o z/OS pode agir como um servidor IBM MQ, ao qual os aplicativos clientes IBM MQ possam se conectar.

**IBM MQ MQI client Grupos de gerenciadores de filas:** se o nome especificado começar com um asterisco (\*), o gerenciador de filas com o qual a conexão é feita poderá ter um nome diferente daquele especificado pelo aplicativo. O nome especificado (sem o asterisco) define um *grupo* de gerenciadores de filas que são elegíveis para conexão. A implementação seleciona um do grupo, tentando um de cada vez, até localizar um que permita uma conexão. A ordem em que as conexões são tentadas é influenciada pelo peso do canal do cliente e pelos valores de afinidade de conexão dos canais candidatos. Se nenhum dos gerenciadores de filas no grupo estiver disponível para conexão, a chamada falhará. Cada gerenciador de filas é tentado somente uma vez. Se um asterisco sozinho for especificado para o nome, um grupo de gerenciadores de filas padrão definido pela implementação será usado.

Os grupos de gerenciadores de filas são suportados apenas para aplicativos em execução em um ambiente do cliente MQ; a chamada falhará se um aplicativo não cliente especificar um nome do gerenciador de filas que comece com um asterisco. Um grupo é definido fornecendo várias definições de canal de conexão do cliente com o mesmo nome do gerenciador de filas (o nome especificado sem o asterisco) para se comunicar com cada um dos gerenciadores de filas no grupo. O grupo padrão é definido fornecendo uma ou mais definições de canal de conexão do cliente, cada uma com um nome de gerenciador de filas em branco (especificar um nome todo em branco, portanto, tem o mesmo efeito que especificar um único asterisco para o nome de um aplicativo cliente).

Depois de se conectar a um gerenciador de filas de um grupo, um aplicativo pode especificar espaços em branco na maneira típica nos campos de nome do gerenciador de filas nos descritores de mensagem e objeto para significar o nome do gerenciador de filas ao qual o aplicativo se conectou (o *gerenciador de filas locais*). Se o aplicativo precisar saber esse nome, use a chamada MQINQ para consultar o atributo do gerenciador de filas do **QMGRName**.

Prefixar um asterisco com o nome da conexão sugere que o aplicativo não dependa da conexão com um determinado gerenciador de filas no grupo. Os aplicativos adequados são:

- Aplicativos que colocam mensagens, mas não obtêm mensagens.
- Aplicativos que colocam mensagens de solicitação e, em seguida, obtêm as mensagens de resposta de uma fila *dinâmica temporária*.

Aplicativos inadequados são aqueles que precisam obter mensagens de uma fila específica em um gerenciador de filas específico; esses aplicativos não devem prefixar o nome com um asterisco.

Se você especificar um asterisco, o comprimento máximo do restante do nome será de 47 caracteres.

O comprimento desse parâmetro é fornecido por MQ\_Q\_MGR\_NAME\_LENGTH.

## Hconn

Tipo: MQHCONN - saída

Esta manipulação representa a conexão ao gerenciador de filas. Especifique-a em todas as chamadas de enfileiramento de mensagens subsequentes emitidas pelo aplicativo. Ela deixa de ser válida quando a chamada MQDISC é emitida ou quando a unidade de processamento que define o escopo da manipulação é finalizada.

O IBM MQ agora fornece a biblioteca mqm com pacotes do cliente, bem como pacotes do servidor. Isso significa que, quando uma chamada MQI localizada na biblioteca mqm é feita, o tipo de conexão é verificado para ver se é uma conexão do cliente ou do servidor e, em seguida, a chamada subjacente correta é feita. Portanto, uma saída para a qual é passada um *Hconn* pode agora ser vinculada à biblioteca mqm, mas usada em uma instalação do cliente.

*Escopo da manipulação:* O escopo da manipulação retornada depende da chamada usada para se conectar ao gerenciador de filas (MQCONN ou MQCONNX). Se a chamada usada for MQCONNX, o escopo do identificador também dependerá da opção MQCNO\_HANDLE\_SHARE\_\* especificada no campo *Options* da estrutura MQCNO.

- Se a chamada for MQCONN ou a opção MQCNO\_HANDLE\_SHARE\_NONE for especificada, a manipulação retornada será uma manipulação *não compartilhada*.

O escopo de uma manipulação não compartilhada é a menor unidade de processamento paralelo suportada pela plataforma na qual o aplicativo está em execução (consulte [Tabela 546 na página 688](#) para obter detalhes); a manipulação não é válida fora da unidade de processamento paralelo da qual a chamada foi emitida.

- Se você especificar a opção MQCNO\_HANDLE\_SHARE\_BLOCK ou MQCNO\_HANDLE\_SHARE\_NO\_BLOCK, a manipulação retornada será uma manipulação *compartilhada*.

O escopo de uma manipulação compartilhada é o processo que possui o encadeamento do qual a chamada foi emitida; a manipulação pode ser usada de qualquer encadeamento pertencente a esse processo. Nem todas as plataformas suportam encadeamentos.

- Se a chamada MQCONN ou MQCONNX falhar com o código de conclusão igual a MQCC\_FAILED, o valor Hconn ficará indefinido.

<b>Plataforma</b>	<b>Escopo da manipulação não compartilhada</b>
z/OS	<ul style="list-style-type: none"><li>• CICS: a tarefa CICS</li><li>• IMS: a tarefa, até o próximo ponto de sincronização (excluindo as subtarefas da tarefa)</li><li>• Lote e TSO do z/OS: a tarefa (excluindo as subtarefas da tarefa)</li></ul>
IBM i	Emprego
AIX and Linux	Encadeamento
Aplicativos Windows de 32 bits	Encadeamento
Aplicativos Windows de 64 bits	Encadeamento

Em aplicativos z/OS for CICS, o valor retornado é:

### **MQHC\_DEF\_HCONN**

Manipulação de conexões padrão.

## CompCode

Tipo: MQLONG - saída

O código de conclusão; é um dos seguintes:

**MQCC\_OK**

Indica conclusão bem-sucedida.

**MQCC\_WARNING**

Aviso (conclusão parcial).

**MQCC\_FAILED**

A chamada falhou.

**Razão**

Tipo: MQLONG - saída

Se *CompCode* for MQCC\_OK:

**MQRC\_NONE**

(0, X'000') Nenhuma razão a relatar.

Se *CompCode* for MQCC\_WARNING:

**MQRC\_ALREADY\_CONNECTED**

(2002, X'7D2') Aplicativo já conectado.

**MQRC\_CLUSTER\_EXIT\_LOAD\_ERROR**

(2267, X'8DB') Não é possível carregar a saída de carga de trabalho do cluster.

**MQRC\_SSL\_ALREADY\_INITIALIZED**

(2391, X'957') SSL já inicializado.

Se *CompCode* for MQCC\_FAILED:

**MQRC\_ADAPTER\_CONN\_LOAD\_ERROR**

(2129, X'851') Não é possível carregar o módulo de conexão do adaptador.

**MQRC\_ADAPTER\_DEFS\_ERROR**

(2131, X'853') O módulo de definição do subsistema do adaptador não é válido.

**MQRC\_ADAPTER\_DEFS\_LOAD\_ERROR**

(2132, X'854') Não é possível carregar o módulo de definição do subsistema do adaptador.

**MQRC\_ADAPTER\_NOT\_AVAILABLE**

(2204, X'89C') Adaptador não disponível.

**MQRC\_ADAPTER\_SERV\_LOAD\_ERROR**

(2130, X'852') Não foi possível carregar o módulo de serviço do adaptador.

**MQRC\_ADAPTER\_STORAGE\_SHORTAGE**

(2127, X'84F') Armazenamento insuficiente para o adaptador.

**MQRC\_ANOTHER\_Q\_MGR\_CONNECTED**

(2103, X'837') Outro gerenciador de filas já conectado.

**MQRC\_API\_EXIT\_ERROR**

(2374, X'946') Falha na saída de API.

**MQRC\_API\_EXIT\_INIT\_ERROR**

(2375, X'947') Falha na inicialização da saída de API.

**MQRC\_API\_EXIT\_TERM\_ERROR**

(2376, X'948') Falha na finalização da saída de API.

**MQRC\_ASID\_MISMATCH**

(2157, X'86D') ASIDs de início e primárias são diferentes.

**MQRC\_BUFFER\_LENGTH\_ERROR**

(2005, X'7D5') Parâmetro de comprimento de buffer inválido.

**MQRC\_CALL\_IN\_PROGRESS**

(2219, X'8AB') Chamada MQI inserida antes da chamada anterior concluída.

**MQRC\_CONN\_ID\_IN\_USE**

(2160, X'870') O identificador de conexão já está em uso.

**MQRC\_CONNECTION\_BROKEN**

(2009, X'7D9') Conexão ao gerenciador de filas perdida.

**MQRC\_CONNECTION\_ERROR**

(2273, X'8E1') Erro ao processar a chamada MQCONN.

**MQRC\_CONNECTION\_NOT\_AVAILABLE**

(2568, X'A08') Ocorre em uma chamada MQCONN ou MQCONNX quando o gerenciador de filas não consegue fornecer uma conexão do tipo de conexão solicitado na instalação atual. Uma conexão do cliente não pode ser feita em uma instalação apenas de servidor. Uma conexão local não pode ser feita em uma instalação somente do cliente.

**MQRC\_CONNECTION QUIESCING**

(2202, X'89A') Quiesce de conexão.

**MQRC\_CONNECTION\_STOPPING**

(2203, X'89B') Conexão sendo encerrada.

**MQRC\_CRYPTO\_HARDWARE\_ERROR**

(2382, X'94E') Erro de configuração de hardware de criptografia.

**MQRC\_DUPLICATE\_RECOV\_COORD**

(2163, X'873') O coordenador de recuperação existe.

**MQRC\_ENVIRONMENT\_ERROR**

(2012, X'7DC') Chamada não válida no ambiente.

Além disso, na chamada MQCONNX, passando o bloco de controle [“MQCSP-Parâmetros de segurança”](#) na [página 343](#) de um aplicativo CICS ou IMS.

**MQRC\_HCONN\_ERROR**

(2018, X'7E2') Manipulação de conexões não válida.

**MQRC\_HOST\_NOT\_AVAILABLE**

(2538, X'9EA') Uma chamada MQCONN foi emitida de um cliente para se conectar a um gerenciador de filas, mas a tentativa de alocar uma conversa para o sistema remoto falhou.

**MQRC\_INSTALLATION\_MISMATCH**

(2583, X'A17') Incompatibilidade entre a instalação do gerenciador de filas e a biblioteca selecionada.

**MQRC\_KEY\_REPOSITORY\_ERROR**

(2381, X'94D') O repositório de chaves não é válido.

**MQRC\_MAX\_CONNS\_LIMIT\_REACHED**

(2025, X'7E9') Atingido o número máximo de conexões.

**MQRC\_NOT\_AUTHORIZED**

(2035, X'7F3') Não autorizado para acesso.

**MQRC\_OPEN\_FAILED**

(2137, X'859') Objeto não aberto com sucesso.

**MQRC\_Q\_MGR\_NAME\_ERROR**

(2058, X'80A') Nome do gerenciador de filas não válido ou não conhecido.

**MQRC\_Q\_MGR\_NOT\_AVAILABLE**

(2059, X'80B') Gerenciador de filas não disponível para conexão.

**MQRC\_Q\_MGR QUIESCING**

(2161, X'871') Quiesce do gerenciador de filas.

**MQRC\_Q\_MGR\_STOPPING**

(2162, X'872') Gerenciador de filas sendo encerrado.

**MQRC\_RESOURCE\_PROBLEM**

(2102, X'836') Recursos insuficientes do sistema disponíveis.

**MQRC\_SECURITY\_ERROR**

(2063, X'80F') Ocorreu um erro na segurança.

**MQRC\_SSL\_INITIALIZATION\_ERROR**

(2393, X'959') Erro de inicialização de SSL.

**MQRC\_STORAGE\_NOT\_AVAILABLE**

(2071, X'817') Armazenamento insuficiente disponível.

**MQRC\_UNEXPECTED\_ERROR**

(2195, X'893') Ocorreu um erro inesperado.


Para obter informações detalhadas sobre esses códigos, consulte [Mensagens e códigos de razão](#).

**Observações de Uso**

1. O gerenciador de filas ao qual a conexão é feita usando a chamada MQCONN é chamado de *gerenciador de filas locais*.
2. As filas pertencentes ao gerenciador de filas locais aparecem para o aplicativo como filas locais. É possível colocar mensagens e obter mensagens dessas filas.  
  
Filas compartilhadas que pertencem ao grupo de filas compartilhadas ao qual o gerenciador de filas locais pertence aparecem para o aplicativo como filas locais. É possível colocar mensagens e obter mensagens dessas filas.  
  
As filas pertencentes a gerenciadores de filas remotas aparecem como filas remotas. É possível colocar mensagens nessas filas, mas não obter mensagens dessas filas.
3. Se o gerenciador de filas falhar enquanto um aplicativo estiver em execução, o aplicativo deverá emitir a chamada MQCONN novamente para obter uma nova manipulação de conexões para usar em chamadas do IBM MQ subsequentes. O aplicativo pode emitir a chamada MQCONN periodicamente, até que ela seja bem-sucedida.  
  
Se um aplicativo não tiver certeza se está conectado ao gerenciador de filas, ele poderá emitir uma chamada MQCONN com segurança para obter uma manipulação de conexões. Se o aplicativo já estiver conectado, a manipulação retornada será a mesma que aquela retornada pela chamada MQCONN anterior, mas com o código de conclusão MQCC\_WARNING e o código de razão MQRC\_ALREADY\_CONNECTED.
4. Quando o aplicativo tiver concluído o uso de chamadas IBM MQ, ele deverá usar a chamada MQDISC para se desconectar do gerenciador de filas.
5. Se a chamada MQCONN falhar com o código de conclusão igual a MQCC\_FAILED, o valor Hconn ficará indefinido.
6. No z/OS:
  - Os aplicativos em lote, TSO e IMS devem emitir a chamada MQCONN para usar as outras chamadas IBM MQ. Esses aplicativos podem se conectar a mais de um gerenciador de filas simultaneamente.  
  
Se o gerenciador de filas falhar, o aplicativo deverá emitir a chamada novamente depois de o gerenciador de filas ter sido reiniciado para obter uma nova manipulação de conexões.  
  
Embora os aplicativos IMS possam emitir a chamada MQCONN repetidamente, mesmo quando já conectados, isso não é recomendado para message processing programs (MPPs) on-line.
  - Os aplicativos CICS não precisam emitir a chamada MQCONN para usar as outras chamadas IBM MQ, mas poderão fazer isso se quiserem; ambas as chamadas, MQCONN e MQDISC, são aceitas. No entanto, não é possível se conectar a mais de um gerenciador de filas simultaneamente.  
  
Se o gerenciador de filas falhar, esses aplicativos serão reconectados automaticamente quando o gerenciador de filas for reiniciado e, portanto, não precisarão emitir a chamada MQCONN.
7. No z/OS, para definir os gerenciadores de filas disponíveis:
  - Para aplicativos em lote, os programadores do sistema podem usar a macro CSQBDEF para criar um módulo (CSQBDEFV) que define o nome do gerenciador de filas padrão ou o nome do grupo de filas compartilhadas..

- Para aplicativos IMS, os programadores de sistema podem usar a macro CSQQDEFX para criar um módulo (CSQQDEFV) que defina os nomes dos gerenciadores de filas disponíveis e especifique o gerenciador de filas padrão.

Além disso, cada gerenciador de filas deve ser definido para a região de controle do IMS e para cada região dependente que estiver acessando esse gerenciador de filas. Para fazer isso, deve-se criar um membro do subsistema na biblioteca IMS.PROCLIB e identificar o membro do subsistema para as regiões aplicáveis do IMS. Se um aplicativo tentar se conectar a um gerenciador de filas não definido no membro do subsistema para sua região IMS, ele será encerrado de forma anormal.

 Para obter mais informações sobre como usar essas macros, veja [Macros destinadas para uso do cliente](#).

8. No IBM i, os programas que terminam de forma anormal não são desconectados automaticamente do gerenciador de filas. Grave aplicativos para permitir a possibilidade de a chamada MQCONN ou MQCONNX retornar o código de conclusão MQCC\_WARNING e o código de razão MQRC\_ALREADY\_CONNECTED. Use a manipulação de conexões retornada nesta situação normalmente.

## Chamada C

```
MQCONN (QMgrName, &Hconn, &CompCode, &Reason);
```

Declare os parâmetros conforme a seguir:

```
MQCHAR48 QMgrName; /* Name of queue manager */
MQHCONN Hconn; /* Connection handle */
MQLONG CompCode; /* Completion code */
MQLONG Reason; /* Reason code qualifying CompCode */
```

## Chamada COBOL

```
CALL 'MQCONN' USING QMGRNAME, HCONN, COMPCODE, REASON.
```

Declare os parâmetros conforme a seguir:

```
** Name of queue manager
01 QMGRNAME PIC X(48).
** Connection handle
01 HCONN PIC S9(9) BINARY.
** Completion code
01 COMPCODE PIC S9(9) BINARY.
** Reason code qualifying COMPCODE
01 REASON PIC S9(9) BINARY.
```

## Chamada PL/I

```
call MQCONN (QMgrName, Hconn, CompCode, Reason);
```

Declare os parâmetros conforme a seguir:

```
dcl QMgrName char(48); /* Name of queue manager */
dcl Hconn fixed bin(31); /* Connection handle */
dcl CompCode fixed bin(31); /* Completion code */
dcl Reason fixed bin(31); /* Reason code qualifying CompCode */
```



## Chamada do Assembler de Alto Nível

```
CALL MQCONN, (QMGRNAME, HCONN, COMPCODE, REASON)
```

Declare os parâmetros conforme a seguir:

QMGRNAME	DS	CL48	Name of queue manager
HCONN	DS	F	Connection handle
COMPCODE	DS	F	Completion code
REASON	DS	F	Reason code qualifying COMPCODE

## Chamada do Visual Basic

```
MQCONN QMgrName, Hconn, CompCode, Reason
```

Declare os parâmetros conforme a seguir:

```
Dim QMgrName As String*48 'Name of queue manager'  
Dim Hconn As Long 'Connection handle'  
Dim CompCode As Long 'Completion code'  
Dim Reason As Long 'Reason code qualifying CompCode'
```

## MQCONNX - Conectar Gerenciador de Filas (Estendido)

A chamada MQCONNX conecta um programa aplicativo a um gerenciador de filas. Ele fornece uma manipulação de conexões do gerenciador de filas, que é usada pelo aplicativo em chamadas subsequentes do IBM MQ.

A chamada MQCONNX é como a chamada MQCONN, exceto que MQCONNX permite que opções sejam especificadas para controlar a maneira como a chamada funciona.

- Essa chamada é suportada em todos os sistemas IBM MQ, e clientes IBM MQ conectados a esses sistemas.

Uma conexão do cliente não pode ser estabelecida em uma instalação de servidor apenas e uma conexão local não pode ser estabelecida em uma instalação de cliente apenas.

### Sintaxe

```
MQCONNX (QMgrName, ConnectOpts, Hconn, CompCode, Reason)
```

### Parâmetros

#### QMgrName

Tipo: MQCHAR48 - entrada

Consulte o parâmetro **QMgrName** descrito em [“MQCONN - Conectar o gerenciador de filas”](#) na página 685 para obter detalhes..

#### ConnectOpts

Tipo: MQCNO-entrada/saída

Consulte [“MQCNO-Opções de conexão”](#) na página 324 para obter detalhes.

#### Hconn

Tipo: MQHCONN - saída

Esta manipulação representa a conexão ao gerenciador de filas. Especifique-a em todas as chamadas de enfileiramento de mensagens subsequentes emitidas pelo aplicativo. Ela deixa de ser válida

quando a chamada MQDISC é emitida ou quando a unidade de processamento que define o escopo da manipulação é finalizada.

O IBM MQ agora fornece a biblioteca mqm com pacotes do cliente, bem como pacotes do servidor. Isso significa que, quando uma chamada MQI localizada na biblioteca mqm é feita, o tipo de conexão é verificado para ver se é uma conexão do cliente ou do servidor e, em seguida, a chamada subjacente correta é feita. Portanto, uma saída para a qual é passada um *Hconn* pode agora ser vinculada à biblioteca mqm, mas usada em uma instalação do cliente.

*Escopo da manipulação:* O escopo da manipulação retornada depende da chamada usada para se conectar ao gerenciador de filas (MQCONN ou MQCONNX). Se a chamada usada for MQCONNX, o escopo do identificador também dependerá da opção MQCNO\_HANDLE\_SHARE\_\* especificada no campo *Options* da estrutura MQCNO.

- Se a chamada for MQCONN ou a opção MQCNO\_HANDLE\_SHARE\_NONE for especificada, a manipulação retornada será uma manipulação *não compartilhada*.

O escopo de uma manipulação não compartilhada é a menor unidade de processamento paralelo suportada pela plataforma na qual o aplicativo está em execução (consulte [Tabela 547 na página 694](#) para obter detalhes); a manipulação não é válida fora da unidade de processamento paralelo da qual a chamada foi emitida.

- Se você especificar a opção MQCNO\_HANDLE\_SHARE\_BLOCK ou MQCNO\_HANDLE\_SHARE\_NO\_BLOCK, a manipulação retornada será uma manipulação *compartilhada*.

O escopo de uma manipulação compartilhada é o processo que possui o encadeamento do qual a chamada foi emitida; a manipulação pode ser usada de qualquer encadeamento pertencente a esse processo. Nem todas as plataformas suportam encadeamentos.

- Se a chamada MQCONN ou MQCONNX falhar com o código de conclusão igual a MQCC\_FAILED, o valor *Hconn* ficará indefinido.

<i>Tabela 547. Escopo de manipulações não compartilhadas em várias plataformas</i>	
<b>Plataforma</b>	<b>Escopo da manipulação não compartilhada</b>
z/OS	<ul style="list-style-type: none"> <li>• CICS: a tarefa CICS</li> <li>• IMS: a tarefa, até o próximo ponto de sincronização (excluindo as subtarefas da tarefa)</li> <li>• Lote e TSO do z/OS: a tarefa (excluindo as subtarefas da tarefa)</li> </ul>
IBM i	Emprego
AIX and Linux	Encadeamento
Aplicativos Windows de 32 bits	Encadeamento
Aplicativos Windows de 64 bits	Encadeamento

Em aplicativos z/OS for CICS, o valor retornado é:

**MQHC\_DEF\_HCONN**

Manipulação de conexões padrão.

**CompCode**

Tipo: MQLONG - saída

Consulte o parâmetro **CompCode** descrito em [“MQCONN - Conectar o gerenciador de filas” na página 685](#) para obter detalhes..

**Razão**

Tipo: MQLONG - saída

Os seguintes códigos podem ser retornados pelas chamadas MQCONN e MQCONNX. Para obter uma lista de códigos adicionais que podem ser retornados pela chamada MQCONNX, consulte os códigos a seguir:

Se *CompCode* for MQCC\_OK:

**MQRC\_NONE**

(0, X'000') Nenhuma razão a relatar.

Se *CompCode* for MQCC\_WARNING:

**MQRC\_ALREADY\_CONNECTED**

(2002, X'7D2') Aplicativo já conectado.

**MQRC\_CLUSTER\_EXIT\_LOAD\_ERROR**

(2267, X'8DB') Não é possível carregar a saída de carga de trabalho do cluster.

**MQRC\_SSL\_ALREADY\_INITIALIZED**

(2391, X'957') SSL já inicializado.

Se *CompCode* for MQCC\_FAILED:

**MQRC\_ADAPTER\_CONN\_LOAD\_ERROR**

(2129, X'851') Não é possível carregar o módulo de conexão do adaptador.

**MQRC\_ADAPTER\_DEFS\_ERROR**

(2131, X'853') O módulo de definição do subsistema do adaptador não é válido.

**MQRC\_ADAPTER\_DEFS\_LOAD\_ERROR**

(2132, X'854') Não é possível carregar o módulo de definição do subsistema do adaptador.

**MQRC\_ADAPTER\_NOT\_AVAILABLE**

(2204, X'89C') Adaptador não disponível.

**MQRC\_ADAPTER\_SERV\_LOAD\_ERROR**

(2130, X'852') Não foi possível carregar o módulo de serviço do adaptador.

**MQRC\_ADAPTER\_STORAGE\_SHORTAGE**

(2127, X'84F') Armazenamento insuficiente para o adaptador.

**MQRC\_ANOTHER\_Q\_MGR\_CONNECTED**

(2103, X'837') Outro gerenciador de filas já conectado.

**MQRC\_API\_EXIT\_ERROR**

(2374, X'946') Falha na saída de API.

**MQRC\_API\_EXIT\_INIT\_ERROR**

(2375, X'947') Falha na inicialização da saída de API.

**MQRC\_API\_EXIT\_TERM\_ERROR**

(2376, X'948') Falha na finalização da saída de API.

**MQRC\_ASID\_MISMATCH**

(2157, X'86D') ASIDs de início e primárias são diferentes.

**MQRC\_BUFFER\_LENGTH\_ERROR**

(2005, X'7D5') Parâmetro de comprimento de buffer inválido.

**MQRC\_CALL\_IN\_PROGRESS**

(2219, X'8AB') Chamada MQI inserida antes da chamada anterior concluída.

**MQRC\_CONN\_ID\_IN\_USE**

(2160, X'870') O identificador de conexão já está em uso.

**MQRC\_CONNECTION\_BROKEN**

(2009, X'7D9') Conexão ao gerenciador de filas perdida.

**MQRC\_CONNECTION\_ERROR**

(2273, X'8E1') Erro ao processar a chamada MQCONN.

**MQRC\_CONNECTION\_NOT\_AVAILABLE**

(2568, X'A08') Ocorre em uma chamada MQCONN ou MQCONNX quando o gerenciador de filas não consegue fornecer uma conexão do tipo de conexão solicitado na instalação atual. Uma

conexão do cliente não pode ser feita em uma instalação apenas de servidor. Uma conexão local não pode ser feita em uma instalação somente do cliente.

**MQRC\_CONNECTION QUIESCING**

(2202, X'89A') Quiesce de conexão.

**MQRC\_CONNECTION STOPPING**

(2203, X'89B') Conexão sendo encerrada.

**MQRC\_CRYPTO\_HARDWARE\_ERROR**

(2382, X'94E') Erro de configuração de hardware de criptografia.

**MQRC\_DUPLICATE\_RECOV\_COORD**

(2163, X'873') O coordenador de recuperação existe.

**MQRC\_ENVIRONMENT\_ERROR**

(2012, X'7DC') Chamada não válida no ambiente.

Além disso, na chamada MQCONNX, passando o bloco de controle “MQCSP-Parâmetros de segurança” na página 343 de um aplicativo CICS ou IMS.

**MQRC\_HCONN\_ERROR**

(2018, X'7E2') Manipulação de conexões não válida.

**MQRC\_HOST\_NOT\_AVAILABLE**

(2538, X'9EA') Uma chamada MQCONN foi emitida de um cliente para se conectar a um gerenciador de filas, mas a tentativa de alocar uma conversa para o sistema remoto falhou.

**MQRC\_INSTALLATION\_MISMATCH**

(2583, X'A17') Incompatibilidade entre a instalação do gerenciador de filas e a biblioteca selecionada.

**MQRC\_KEY\_REPOSITORY\_ERROR**

(2381, X'94D') O repositório de chaves não é válido.

**MQRC\_MAX\_CONNS\_LIMIT\_REACHED**

(2025, X'7E9') Atingido o número máximo de conexões.

**MQRC\_NOT\_AUTHORIZED**

(2035, X'7F3') Não autorizado para acesso.

**MQRC\_OPEN\_FAILED**

(2137, X'859') Objeto não aberto com sucesso.

**MQRC\_Q\_MGR\_NAME\_ERROR**

(2058, X'80A') Nome do gerenciador de filas não válido ou não conhecido.

**MQRC\_Q\_MGR\_NOT\_AVAILABLE**

(2059, X'80B') Gerenciador de filas não disponível para conexão.

**MQRC\_Q\_MGR QUIESCING**

(2161, X'871') Quiesce do gerenciador de filas.

**MQRC\_Q\_MGR STOPPING**

(2162, X'872') Gerenciador de filas sendo encerrado.

**MQRC\_RESOURCE\_PROBLEM**

(2102, X'836') Recursos insuficientes do sistema disponíveis.

**MQRC\_SECURITY\_ERROR**

(2063, X'80F') Ocorreu um erro na segurança.

**MQRC\_SSL\_INITIALIZATION\_ERROR**

(2393, X'959') Erro de inicialização de SSL.

**MQRC\_STORAGE\_NOT\_AVAILABLE**

(2071, X'817') Armazenamento insuficiente disponível.

**MQRC\_UNEXPECTED\_ERROR**

(2195, X'893') Ocorreu um erro inesperado.

Os seguintes códigos de razão adicionais podem ser retornados pela chamada MQCONNX:

Se *CompCode* for MQCC\_FAILED:

**MQRC\_AIR\_ERROR**

(2385, X' 951 ') Registro de informações sobre autenticação inválido.

**MQRC\_AUTH\_INFO\_CONN\_NAME\_ERROR**

(2387, X' 953 ') Nome da conexão de informações sobre autenticação inválido.

**MQRC\_AUTH\_INFO\_REC\_COUNT\_ERROR**

(2383, X'94F') Contagem de registro de informações sobre autenticação não válida.

**MQRC\_AUTH\_INFO\_REC\_ERROR**

(2384, X' 950 ') Campos de registro de informações sobre autenticação inválidos.

**MQRC\_AUTH\_INFO\_TYPE\_ERROR**

(2386, X' 952 ') Tipo de informação de autenticação inválido.

**MQRC\_CD\_ERROR**

(2277, X'8E5') Definição de canal não válida.

**MQRC\_CLIENT\_CONN\_ERROR**

(2278, X'8E6') Campos de conexão do cliente não válidos.

**MQRC\_CNO\_ERROR**

(2139, X'85B') Estrutura de opções de conexão não válida.

**MQRC\_CONN\_TAG\_IN\_USE**

(2271, X'8DF') Tag de conexão em uso.

**MQRC\_CONN\_TAG\_NOT\_USABLE**

(2350, X'92E') Tag de conexão não utilizável.

**MQRC\_CSP\_ERROR**

(2595, X'A23') A estrutura MQCSP não é válida.

**V 9.3.4 MQRC\_FUNCTION\_NOT\_SUPPORTED**

(2298, X'8FA') A função solicitada não está disponível no ambiente atual.

**MQRC\_LDAP\_PASSWORD\_ERROR**

(2390, X' 956 ') A senha LDAP não é válida.

**MQRC\_LDAP\_USER\_NAME\_ERROR**

(2388, X' 954 ') Os campos de nome de usuário LDAP não são válidos.

**MQRC\_LDAP\_USER\_NAME\_LENGTH\_ERR**

(2389, X' 955 ') O comprimento do nome do usuário LDAP não é válido.

**MQRC\_OPTIONS\_ERROR**

(2046, X'7FE') Opções não válidas ou não consistentes.

**MQRC\_SCO\_ERROR**

(2380, X'94C') Estrutura de opções de configuração SSL não válida.

**MQRC\_SSL\_CONFIG\_ERROR**

(2392, X' 958 ') Erro de configuração SSL.

**MQRC\_TOKEN\_TIMESTAMP\_NOT\_VALID**

(2064, X'810 ') O token de autenticação ainda não é válido ou está expirado

Para obter informações detalhadas sobre esses códigos, consulte [Mensagens e códigos de razão](#).

## Observações de Uso

Para a linguagem de programação Visual Basic, aplica-se o seguinte ponto:

- O parâmetro **ConnectOpts** é declarado como sendo do tipo MQCNO Se o aplicativo estiver em execução como um IBM MQ MQI cliente você deseja especificar os parâmetros do canal de conexão do cliente, declare o parâmetro **ConnectOpts** como sendo do tipo Any, para que o aplicativo possa especificar uma estrutura MQCNOCD na chamada no lugar de uma estrutura MQCNO. No entanto, isso significa que o parâmetro **ConnectOpts** não pode ser verificado para assegurar que ele seja o tipo de dados correto

## Chamada C

```
MQCONN (QMgrName, &ConnectOpts, &Hconn, &CompCode, &Reason);
```

Declare os parâmetros conforme a seguir:

```
MQCHAR48 QMgrName; /* Name of queue manager */
MQCNO ConnectOpts; /* Options that control the action of MQCONN */
MQHCONN Hconn; /* Connection handle */
MQLONG CompCode; /* Completion code */
MQLONG Reason; /* Reason code qualifying CompCode */
```

## Chamada COBOL

```
CALL 'MQCONN' USING QMGRNAME, CONNECTOPTS, HCONN, COMPCODE,
REASON.
```

Declare os parâmetros conforme a seguir:

```
** Name of queue manager
01 QMGRNAME PIC X(48).
** Options that control the action of MQCONN
01 CONNECTOPTS.
COPY CMQCNOV.
** Connection handle
01 HCONN PIC S9(9) BINARY.
** Completion code
01 COMPCODE PIC S9(9) BINARY.
** Reason code qualifying COMPCODE
01 REASON PIC S9(9) BINARY.
```

## Chamada PL/I

```
call MQCONN (QMgrName, ConnectOpts, Hconn, CompCode, Reason);
```

Declare os parâmetros conforme a seguir:

```
dcl QMgrName char(48); /* Name of queue manager */
dcl ConnectOpts like MQCNO; /* Options that control the action of
MQCONN */
dcl Hconn fixed bin(31); /* Connection handle */
dcl CompCode fixed bin(31); /* Completion code */
dcl Reason fixed bin(31); /* Reason code qualifying CompCode */
```

## Chamada do Assembler de Alto Nível

```
CALL MQCONN,(QMGRNAME,CONNECTOPTS,HCONN,COMPCODE,REASON)
```

Declare os parâmetros conforme a seguir:

```
QMGRNAME DS CL48 Name of queue manager
CONNECTOPTS CMQCNOA , Options that control the action of MQCONN
HCONN DS F Connection handle
COMPCODE DS F Completion code
REASON DS F Reason code qualifying COMPCODE
```

## Chamada do Visual Basic

```
MQCONN, QMgrName, ConnectOpts, Hconn, CompCode, Reason
```

Declare os parâmetros conforme a seguir:

```
Dim QMgrName As String*48 'Name of queue manager'  
Dim ConnectOpts As MQCNO 'Options that control the action of'  
                        'MQCONNX'  
Dim Hconn As Long 'Connection handle'  
Dim CompCode As Long 'Completion code'  
Dim Reason As Long 'Reason code qualifying CompCode'
```

## MQCRTMH-Criar identificador de mensagens

A chamada de MQCRTMH retorna uma manipulação de mensagens

Um aplicativo pode usar a chamada MQCRTMH em chamadas de enfileiramento de mensagens subsequentes:

- Use a chamada [MQSETMP](#) para configurar uma propriedade do identificador de mensagem
- Use a chamada [MQINQMP](#) para consultar o valor de uma propriedade do identificador de mensagem
- Use a chamada [MQDLTMP](#) para excluir uma propriedade da manipulação de mensagens

O identificador de mensagem pode ser usado nas chamadas MQPUT e MQPUT1 para associar as propriedades do identificador de mensagem com aquelas da mensagem que está sendo colocada. Da mesma forma, ao especificar um identificador de mensagens na chamada MQGET, as propriedades da mensagem que está sendo recuperada podem ser acessadas usando o identificador de mensagens quando a chamada MQGET for concluída.

Use [MQDLTMH](#) para excluir o identificador da mensagem.

## Sintaxe

MQCRTMH (*Hconn*, *CrtMsgHOpts*, *Hmsg*, *CompCode*, *Razão*)

## Parâmetros

### Hconn

Tipo: MQHCONN - entrada

Esta manipulação representa a conexão ao gerenciador de filas. O valor de *Hconn* foi retornado por uma chamada anterior MQCONN ou MQCONNX. Se a conexão com o gerenciador de filas deixar de ser válida e nenhuma chamada IBM MQ estiver operando na manipulação de mensagens, [MQDLTMH](#) será chamado implicitamente para excluir a mensagem.

Como alternativa, é possível especificar o seguinte valor:

### MQHC\_UNASSOCIATED\_HCONN

A manipulação de conexões não representa uma conexão com nenhum gerenciador de fila específico.

Quando esse valor for usado, o identificador de mensagens deverá ser excluído com uma chamada explícita para [MQDLTMH](#) para liberar qualquer armazenamento alocado para ele. IBM MQ nunca exclui implicitamente o identificador de mensagem..

Deve haver pelo menos uma conexão válida para um gerenciador de filas estabelecido no encadeamento que está criando o identificador de mensagens, caso contrário, a chamada falhará com MQRC\_HCONN\_ERROR.

Em um ambiente com várias instalações em um único sistema, o valor MQHC\_UNASSOCIATED\_HCONN é limitado para ser usado com a primeira instalação carregada.

no processo.. O código de razão MQRC\_HMSG\_NOT\_AVAILABLE será retornado se a manipulação de mensagem for fornecida para uma instalação diferente

Em z/OS para aplicativos CICS , a chamada MQCONN pode ser omitida e é possível especificar o seguinte valor para *Hconn*:

#### **MQHC\_DEF\_CONN**

Identificador de conexão padrão

#### **CrtMsgHOpts**

Tipo: MQCMHO-entrada

As opções que controlam a ação de MQCRTMH Consulte [MQCMHO](#) para obter detalhes..

#### **Hmsg**

Tipo: MQHMSG-saída

Na saída, é retornada uma manipulação de mensagem que pode ser usada para configurar, consultar e excluir propriedades da manipulação de mensagens Inicialmente, a manipulação de mensagem não contém propriedades

Um identificador de mensagens também possui um descritor de mensagens associado Inicialmente, ele contém os valores padrão Os valores dos campos do descritor de mensagens associados podem ser configurados e consultados usando as chamadas MQSETMP e MQINQMP A chamada MQDLTMP reconfigura um campo do descritor de mensagens para seu valor padrão.

Se o parâmetro *Hconn* for especificado como o valor MQHC\_UNASSOCIATED\_HCONN, a manipulação de mensagem retornada poderá ser usada em chamadas MQGET, MQPUT ou MQPUT1 com qualquer conexão dentro da unidade de processamento, mas poderá estar em uso somente por uma chamada IBM MQ por vez. Se o identificador estiver em uso quando uma segunda chamada IBM MQ tentar usar o mesmo identificador de mensagens, a segunda chamada IBM MQ falhará com o código de razão MQRC\_MSG\_HANDLE\_IN\_USE

Se o parâmetro *Hconn* não for MQHC\_UNASSOCIATED\_HCONN, a manipulação de mensagem retornada poderá ser usada apenas na conexão especificada.

O mesmo valor de parâmetro *Hconn* deve ser usado nas chamadas MQI subsequentes em que essa manipulação de mensagens é usada:

- MQDLTMH
- MQSETMP
- MQINQMP
- MQDLTMP
- MQMHBUF
- MQBUFMH

A manipulação de mensagem retornada deixa de ser válida quando a chamada MQDLTMH é emitida para a manipulação de mensagens ou quando a unidade de processamento que define o escopo da manipulação termina. MQDLTMH será chamado implicitamente se uma conexão específica for fornecida quando a manipulação de mensagens for criada e a conexão com o gerenciador de fila deixar de ser válida, por exemplo, se o MQDBC for chamado

#### **CompCode**

Tipo: MQLONG - saída

O código de conclusão; é um dos seguintes:

#### **MQCC\_OK**

Indica conclusão bem-sucedida.

#### **MQCC\_FAILED**

A chamada falhou.

#### **Razão**

Tipo: MQLONG - saída



Se *CompCode* for MQCC\_OK:

**MQRC\_NONE**

(0, X'000') Nenhuma razão a relatar.

Se *CompCode* for MQCC\_FAILED:

**MQRC\_ADAPTER\_NOT\_AVAILABLE**

(2204, X'089C') Adaptador não disponível.

**MQRC\_ADAPTER\_SERV\_LOAD\_ERROR**

(2130, X'852') Não foi possível carregar o módulo de serviço do adaptador.

**MQRC\_ASID\_MISMATCH**

(2157, X'86D') ASIDs de início e primárias são diferentes.

**MQRC\_CALL\_IN\_PROGRESS**

(2219, X'08AB') chamada MQI inserida antes da chamada anterior ser concluída.

**MQRC\_CMHO\_ERROR**

(2461, X'099D') A estrutura de opções de manipulação de mensagens não é válida.

**MQRC\_CONNECTION\_BROKEN**

(2273, X'7D9') Conexão com o gerenciador de filas perdida.

**MQRC\_HANDLE\_NOT\_AVAILABLE**

(2017, X'07E1') Mais nenhum identificador disponível.

**MQRC\_HCONN\_ERROR**

(2018, X'7E2') Manipulação de conexões não válida.

**MQRC\_HMSG\_ERROR**

(2460, X'099C') Ponteiro do identificador de mensagens inválido.

**MQRC\_OPTIONS\_ERROR**

(2046, X'07FE') Opções não válidas ou não consistentes.

**MQRC\_STORAGE\_NOT\_AVAILABLE**

(2071, X'817') Armazenamento insuficiente disponível.

**MQRC\_UNEXPECTED\_ERROR**

(2195, X'893') Ocorreu um erro inesperado.

Para obter informações detalhadas sobre esses códigos, consulte [Mensagens e códigos de razão](#).

## C

```
MQCRTMH (Hconn, &CrtMsgHOpts, &Hmsg, &CompCode, &Reason);
```

Declare os parâmetros conforme a seguir:

```
MQHCONN  Hconn;          /* Connection handle */
MQCMHO   CrtMsgHOpts;   /* Options that control the action of MQCRTMH */
MQHMSG   Hmsg;          /* Message handle */
MQQLONG  CompCode;     /* Completion code */
MQQLONG  Reason;       /* Reason code qualifying CompCode */
```

## COBOL

```
CALL 'MQCRTMH' USING HCONN, CRTMSGHOPTS, HMSG, COMPCODE, REASON.
```

Declare os parâmetros conforme a seguir:

```
** Connection handle
01 HCONN PIC S9(9) BINARY.
** Options that control the action of MQCRTMH
```

```

01 CRTMSGHOPTS.
   COPY CMQCMHOV.
** Message handle
01 HMSG PIC S9(18) BINARY.
** Completion code
01 COMPCODE PIC S9(9) BINARY.
** Reason code qualifying COMPCODE
01 REASON PIC S9(9) BINARY.

```

## PL/I

```
call MQCRTMH (Hconn, CrtMsgHOpts, Hmsg, CompCode, Reason);
```

Declare os parâmetros conforme a seguir:

```

dcl Hconn          fixed bin(31); /* Connection handle */
dcl CrtMsgHOpts   like MQCMHO; /* Options that control the action of MQCRTMH */
dcl Hmsg          fixed bin(63); /* Message handle */
dcl CompCode      fixed bin(31); /* Completion code */
dcl Reason        fixed bin(31); /* Reason code qualifying CompCode */

```

## High Level Assembler

```
CALL MQCRTMH, (HCONN, CRTMSGHOPTS, HMSG, COMPCODE, REASON)
```

Declare os parâmetros conforme a seguir:

```

HCONN          DS          F      Connection handle
CRTMSGHOPTS    DS          CMQCMHOA , Options that control the action of MQCRTMH
HMSG           DS          D      Message handle
COMPCODE       DS          F      Completion code
REASON         DS          F      Reason code qualifying COMPCODE

```

## MQCTL-Retornos de chamada de controle

A chamada MQCTL executa ações de controle em retornos de chamadas e os identificadores de objetos abertos para uma conexão

### Sintaxe

MQCTL (*Hconn*, *Operação ControlOpts*, *CompCode*, *Motivo*)

### Parâmetros

#### Hconn

Tipo: MQHCONN - entrada

Esta manipulação representa a conexão ao gerenciador de filas. O valor de *Hconn* foi retornado por uma chamada anterior MQCONN ou MQCONNX.

No z/OS para aplicativos CICS, a chamada MQCONN pode ser omitida e é possível especificar o valor especial a seguir para *Hconn*:

#### MQHC\_DEF\_HCONN

Manipulação de conexões padrão.

#### Operação

Tipo: MQLONG - entrada

A operação sendo processada no retorno de chamada definido para o identificador de objeto especificado. Deve-se especificar uma e apenas uma das opções a seguir:

## **MQOP\_START**

Iniciar o consumo de mensagens para todas as funções de consumidor de mensagens definidas para o identificador de conexão especificado

Retornos de chamada executados em um encadeamento iniciado pelo sistema, que é diferente de qualquer um dos encadeamentos de aplicativos.

Esta operação fornece o controle do identificador de conexão fornecido para o sistema. As únicas chamadas MQI que podem ser emitidas por um encadeamento diferente do encadeamento do consumidor são:

- MQCTL com Operação MQOP\_STOP
- MQCTL com Operação MQOP\_SUSPEND
- MQDISC-Executa MQCTL com Operação MQOP\_STOP antes de desconectar o HConn.

MQRC\_HCONN\_ASYNC\_ACTIVE será retornado se uma chamada API IBM MQ for emitida enquanto a manipulação de conexões for iniciada e a chamada não for originada de uma função do consumidor de mensagem..

Se um consumidor de mensagem parar a conexão durante o MQCBCT\_START\_CALL, a chamada MQCTL retornará com um código de razão de falha MQRC\_CONNECTION\_STOPPED.

Isso pode ser emitido em uma função de consumidor. Para a mesma conexão que a rotina de retorno de chamada, seu único propósito é cancelar uma operação MQOP\_STOP emitida anteriormente.

Esta opção não é suportada nos seguintes ambientes: CICS on z/OS ou se o aplicativo estiver ligado a uma biblioteca IBM MQ não encadeada.

## **MQOP\_START\_WAIT**

Iniciar o consumo de mensagens para todas as funções de consumidor de mensagens definidas para o identificador de conexão especificado

Os consumidores de mensagens são executados no mesmo encadeamento e o controle não é retornado ao responsável pela chamada de MQCTL até que:

- Liberado pelo uso das operações MQCTL MQOP\_STOP ou MQOP\_SUSPEND ou
- Todas as rotinas do consumidor tiveram o registro removido ou suspenso.

Se todos os consumidores tiverem o registro removido ou suspenso, uma operação MQOP\_STOP implícita será emitida.

Esta opção não pode ser utilizada a partir de uma rotina de retorno de chamada, seja para o identificador de conexão atual ou qualquer outro identificador de conexão. Se a chamada for tentada, ela retorna com MQRC\_ENVIRONMENT\_ERROR.

Se, a qualquer momento durante uma operação MQOP\_START\_WAIT não houver consumidores registrados, não suspensos, a chamada falhará com um código de razão MQRC\_NO\_CALLBACKS\_ACTIVE.

Se, durante uma operação MQOP\_START\_WAIT, a conexão for suspensa, a chamada MQCTL retorna um código de razão de aviso de MQRC\_CONNECTION\_SUSPENDED; a conexão permanece 'iniciada'.

O aplicativo pode escolher emitir MQOP\_STOP ou MQOP\_RESUME. Nessa instância a operação MQOP\_RESUME bloqueia.

Essa opção não é suportada em um cliente de encadeamento único..

## **MQOP\_STOP**

Pare o consumo de mensagens e aguarde todos os consumidores concluírem suas operações antes que essa opção seja concluída. Esta operação libera o identificador da conexão

Se emitido a partir de uma rotina de retorno de chamada, essa opção não entrará em vigor até que a rotina seja encerrada. Mais nenhuma rotina do consumidor de mensagens é chamada depois que

as rotinas do consumidor para mensagens já lidas foram concluídas e depois que as chamadas de parada (se solicitadas) para rotinas de retorno de chamada foram feitas.

Se emitido fora de uma rotina de retorno de chamada, o controle não retornará ao responsável pela chamada até que as rotinas do consumidor para mensagens já lidas tenham sido concluídas e após chamadas de parada (se solicitadas) para retornos de chamada terem sido feitas. Os retornos de chamada, no entanto, permanecem registrados.

Esta função não tem efeito sobre as mensagens de leitura antecipada. Deve-se assegurar que os consumidores executem MQCLOSE (MQCO\_QUIESCE), a partir da função de retorno de chamada, para determinar se há mensagens adicionais disponíveis para serem entregues

### **MQOP\_SUSPEND**

Pausar o consumo de mensagens Esta operação libera o identificador da conexão

Isso não tem nenhum efeito sobre a leitura antecipada de mensagens para o aplicativo Se você pretende parar o consumo de mensagens por um longo tempo, considere fechar a fila e reabri-la quando o consumo continuar

Se emitido a partir de uma rotina de retorno de chamada, ele não entrará em vigor até que a rotina seja encerrada Mais nenhuma rotina do consumidor de mensagens será chamada após a saída da rotina atual

Se emitido fora de um retorno de chamada, o controle não retornará ao responsável pela chamada até que a rotina do consumidor atual tenha sido concluída e nenhum outro seja chamado.

### **MQOP\_RESUME**

Retome o consumo de mensagens

Essa opção é normalmente emitida a partir do encadeamento do aplicativo principal, mas também pode ser usada a partir de uma rotina de retorno de chamada para cancelar um pedido de suspensão anterior emitido na mesma rotina

Se o MQOP\_RESUME for usado para continuar um MQOP\_START\_WAIT, os blocos de operação.

### **ControlOpts**

Tipo: MQCTLO-entrada

Opções que controlam a ação de MQCTL

Consulte [MQCTLO](#) para obter detalhes da estrutura..

### **CompCode**

Tipo: MQLONG - saída

O código de conclusão; é um dos seguintes:

#### **MQCC\_OK**

Indica conclusão bem-sucedida.

#### **MQCC\_WARNING**

Aviso (conclusão parcial).

#### **MQCC\_FAILED**

A chamada falhou.

### **Razão**

Tipo: MQLONG - saída

Se *CompCode* for MQCC\_OK:

#### **MQRC\_NONE**

(0, X'000') Nenhuma razão a relatar.

Se *CompCode* for MQCC\_FAILED:

#### **MQRC\_ADAPTER\_CONV\_LOAD\_ERROR**

(2133, X'855') Impossível carregar módulos de serviços de conversão de dados.

**MQR\_ADAPTER\_NOT\_AVAILABLE**

(2204, X'89C') Adaptador não disponível.

**MQR\_ADAPTER\_SERV\_LOAD\_ERROR**

(2130, X'852') Não foi possível carregar o módulo de serviço do adaptador.

**MQR\_API\_EXIT\_ERROR**

(2374, X'946') Falha na saída de API.

**MQR\_API\_EXIT\_LOAD\_ERROR**

(2183, X'887') Não foi possível carregar a saída de API.

**MQR\_ASID\_MISMATCH**

(2157, X'86D') ASIDs de início e primárias são diferentes.

**MQR\_BUFFER\_LENGTH\_ERROR**

(2005, X'7D5') Parâmetro de comprimento de buffer inválido.

**MQR\_CALLBACK\_LINK\_ERROR**

(2487, X'9B7') Não é possível chamar a rotina de retorno de chamada

**MQR\_CALLBACK\_NOT\_REGISTRADO**

(2448, X' 990 ') Não é possível remover o registro, suspender ou continuar porque não há retorno de chamada registrado

**MQR\_CALLBACK\_ROUTINE\_ERROR**

(2486, X'9B6') Ambos CallbackFunction e CallbackName foram especificados em uma chamada MQOP\_REGISTER.

Ou CallbackFunction ou CallbackName foi especificado, mas não corresponde à função de retorno de chamada atualmente registrada..

**MQR\_CALLBACK\_TYPE\_ERROR**

(2483, X'9B3') Campo de tipo CallBackincorreto

**MQR\_CALL\_IN\_PROGRESS**

(2219, X'8AB') Chamada MQI inserida antes da chamada anterior concluída.

**MQR\_CBD\_ERROR**

(2444, X'98C') O bloco de opções está incorreto

**MQR\_CBD\_OPTIONS\_ERROR**

(2484, X'9B4') Campo de opções MQCBD incorreto.

**MQR\_CICS\_WAIT\_FAILED**

(2140, X'85C') Pedido de espera rejeitado por CICS.

**MQR\_CONNECTION\_BROKEN**

(2009, X'7D9') Conexão ao gerenciador de filas perdida.

**MQR\_CONNECTION\_NOT\_AUTHORIZED**

(2217, X'8A9') Não autorizado para conexão.

**MQR\_CONNECTION QUIESCING**

(2202, X'89A') Quiesce de conexão.

**MQR\_CONNECTION\_STOPPING**

(2203, X'89B') Conexão sendo encerrada.

**MQR\_CORREL\_ID\_ERROR**

(2207, X'89F') Erro de identificador de correlação.

**MQR\_ENVIRONMENT\_ERROR**

(2012, X'7DC') Chamada não válida no ambiente.

**MQR\_FUNCTION\_NOT\_SUPPORTED**

(2298, X'8FA') A função solicitada não está disponível no ambiente atual.

**MQR\_GET\_INHIBITED**

(2016, X'7E0 ') É inibida para a fila.

**MQR\_GLOBAL\_UOW\_CONFLICT**

(2351, X'92F') Unidades de trabalho globais conflitam.

**MQRC\_GMO\_ERROR**  
(2186, X'88A') Estrutura de opções de obtenção de mensagem inválida.

**MQRC\_HANDLE\_IN\_USE\_FOR\_UOW**  
(2353, X'931') Manipulação em uso para unidade de trabalho global.

**MQRC\_HCONN\_ERROR**  
(2018, X'7E2') Manipulação de conexões não válida.

**MQRC\_HOBJ\_ERROR**  
(2019, X'7E3') Manipulação de objetos não válida.

**MQRC\_INCONSISTENT\_BROWSE**  
(2259, X'8D3') Especificação de pesquisa inconsistente.

**MQRC\_INCONSISTENT\_UOW**  
(2245, X'8C5') Especificação de unidade de trabalho inconsistente.

**MQRC\_INVALID\_MSG\_UNDER\_CURSOR**  
(2246, X'8C6') Mensagem sob o cursor inválida para recuperação.

**MQRC\_LOCAL\_UOW\_CONFLICT**  
(2352, X'930') Unidade de trabalho global conflita com unidade de trabalho local.

**MQRC\_MATCH\_OPTIONS\_ERROR**  
(2247, X'8C7') Opções correspondentes inválidas.

**MQRC\_MAX\_MSG\_LENGTH\_ERROR**  
(2485, X'9B5') Campo de comprimento MaxMsg incorreto

**MQRC\_MD\_ERROR**  
(2026, X'7EA') Descritor de mensagem inválido.

**MQRC\_MODULE\_ENTRY\_NOT\_FOUND**  
(2497, X'9C1') O ponto de entrada da função especificada não pôde ser localizado no módulo.

**MQRC\_MODULE\_INVALID**  
(2496, X'9C0') O módulo foi localizado, mas é do tipo errado (32 bit/64 bit) ou não é uma dll válida.

**MQRC\_MODULE\_NOT\_FOUND**  
(2495, X'9BF') Módulo não localizado no caminho da procura ou não autorizado a carregar.

**MQRC\_MSG\_ID\_ERROR**  
(2206, X'89E') Erro de identificador de mensagem.

**MQRC\_MSG\_SEQ\_NUMBER\_ERROR**  
(2250, X'8CA') Número de sequência da mensagem inválido.

**MQRC\_MSG\_TOKEN\_ERROR**  
(2331, X'91B') Uso de token da mensagem inválido.

**MQRC\_NOT\_OPEN\_FOR\_BROWSE**  
(2036, X'7F4') Fila não aberta para navegação.

**MQRC\_NOT\_OPEN\_FOR\_INPUT**  
(2037, X'7F5') Fila não aberta para entrada.

**MQRC\_OBJECT\_CHANGED**  
(2041, X'7F9') Definição de objeto alterada desde a abertura.

**MQRC\_OBJECT\_DAMAGED**  
(2101, X'835') Objeto danificado.

**MQRC\_OPERATION\_ERROR**  
(2488, X'9B8') Código de operação incorreto na chamada API

**MQRC\_OPTIONS\_ERROR**  
(2046, X'7FE') Opções não válidas ou não consistentes.

**MQRC\_PAGESET\_ERROR**  
(2193, X'891') Erro ao acessar o conjunto de dados do conjunto de páginas.

**MQRC\_Q\_DELETED**

(2052, X'804') A fila foi excluída.

**MQRC\_Q\_INDEX\_TYPE\_ERROR**

(2394, X'95A') Fila possui tipo de índice errado.

**MQRC\_Q\_MGR\_NAME\_ERROR**

(2058, X'80A') Nome do gerenciador de filas não válido ou não conhecido.

**MQRC\_Q\_MGR\_NOT\_AVAILABLE**

(2059, X'80B') Gerenciador de filas não disponível para conexão.

**MQRC\_Q\_MGR QUIESCING**

(2161, X'871') Quiesce do gerenciador de filas.

**MQRC\_Q\_MGR\_STOPPING**

(2162, X'872') Gerenciador de filas sendo encerrado.

**MQRC\_RESOURCE\_PROBLEM**

(2102, X'836') Recursos insuficientes do sistema disponíveis.

**MQRC\_SIGNAL\_OUTSTANDING**

(2069, X'815') Sinais pendentes para essa manipulação.

**MQRC\_STORAGE\_NOT\_AVAILABLE**

(2071, X'817') Armazenamento insuficiente disponível.

**MQRC\_SUPPRESSED\_BY\_EXIT**

(2109, X'83D') Chamada suprimida pelo programa de saída.

**MQRC\_SYNCPOINT\_NOT\_AVAILABLE**

(2072, X'818 ') Suporte de ponto de sincronização não disponível.

**MQRC\_UNEXPECTED\_ERROR**

(2195, X'893') Ocorreu um erro inesperado.

**MQRC\_UOW\_ENLISTMENT\_ERROR**

(2354, X'932') Cadastramento na unidade de trabalho global falhou.

**MQRC\_UOW\_MIX\_NOT\_SUPPORTED**

(2355, X'933') Combinação de chamadas de unidade de trabalho não suportada.

**MQRC\_UOW\_NOT\_AVAILABLE**

(2255, X'8CF') Unidade de trabalho não disponível para o gerenciador de filas a usar.

**MQRC\_WAIT\_INTERVAL\_ERROR**

(2090, X'82A') Intervalo de Espera em MQGMO inválido.

**MQRC\_WRONG\_GMO\_VERSION**

(2256, X'8D0') Versão errada do MQGMO fornecido.

**MQRC\_WRONG\_MD\_VERSION**

(2257, X'8D1') Versão errada do MQMD fornecido.

Para obter informações detalhadas sobre esses códigos, consulte [Mensagens e códigos de razão](#).



## Observações de Uso

1. As rotinas de retorno de chamada devem verificar as respostas de todos os serviços que elas chamam e, se a rotina detectar uma condição que não pode ser resolvida, ela deverá emitir um comando MQCB MQOP\_DEREGISTER para evitar chamadas repetidas à rotina de retorno de chamada.
2. Se estiver usando o consumo assíncrono em um aplicativo no qual um Gerenciador de Transações XA está gerenciando transações globais, incluindo atualizações para o IBM MQ, será necessário considerar os seguintes pontos adicionais:
  - a. Não é válido chamar MQCTL (MQOP\_START) para um **HConn**, após ele ter sido criado, após chamar **xa\_open**.

A razão é que o **HConn** foi anexado a um contexto XA e, portanto, não pode ser acessado no encadeamento separado ou encadeamentos, em uso pelo mecanismo de consumo assíncrono.

- b. Se você chama MQCTL (MQOP\_START) nesse cenário a chamada falha com o código de razão MQRC\_ASYNC\_XA\_CONFLICT (2350).
- c. É válido chamar MQCTL (MQOP\_START\_WAIT) para um **HConn**, depois de ter sido criado, depois de chamar **xa\_open**.

A razão é que esse método de iniciar o mecanismo de consumo assíncrono faz com que todos os retornos de chamada adicionais para o **HConn** sejam executados no encadeamento no qual a chamada MQCTL é feita. Portanto, o link entre **HConn** e o encadeamento não é perdido.

3.  No z/OS, quando a operação for MQOP\_START:
  - Programas que usam rotinas de retorno de chamada assíncronas devem estar autorizados a usar z/OS UNIX System Services (z/OS UNIX).
  - Os programas Language Environment (LE) que usam rotinas de retorno de chamada assíncronas devem usar a opção de tempo de execução LE POSIX(ON).
  - Programas não LE que usam rotinas de retorno de chamada assíncrona não devem usar a interface z/OS UNIX pthread\_create (serviço de chamada BPX1PTC).
4.  MQCTL não é suportado no adaptador IMS .

**Nota:** No CICS, MQOP\_START não é suportado.. Em vez disso, use a chamada de função MQOP\_START\_WAIT

## Chamada C

```
MQCTL (Hconn, Operation, &ControlOpts, &CompCode, &Reason)
```

Declare os parâmetros conforme a seguir:

```
MQHCONN  Hconn;          /* Connection handle */
MQLONG   Operation;     /* Operation being processed */
MQCTLO   ControlOpts    /* Options that control the action of MQCTL */
MQLONG   CompCode;      /* Completion code */
MQLONG   Reason;        /* Reason code qualifying CompCode */
```

## Chamada COBOL

```
CALL 'MQCTL' USING HCONN, OPERATION, CTLOPTS, COMPCODE, REASON.
```

Declare os parâmetros conforme a seguir:

```
** Connection handle
01 HCONN PIC S9(9) BINARY.
** Operation
01 OPERATION PIC S9(9) BINARY.
** Control Options
01 CTLOPTS.
   COPY CMQCTLOV.
** Completion code
01 COMPCODE PIC S9(9) BINARY.
** Reason code qualifying COMPCODE
01 REASON PIC S9(9) BINARY.
```

## Chamada PL/I

```
call MQCTL(Hconn, Operation, CtlOpts, CompCode, Reason)
```

Declare os parâmetros conforme a seguir:



```

dcl Hconn          fixed bin(31); /* Connection handle */
dcl Operation     fixed bin(31); /* Operation */
dcl CtlOpts like  MQCTLO;        /* Options that control the action of MQCTL */
dcl CompCode      fixed bin(31); /* Completion code */
dcl Reason        fixed bin(31); /* Reason code qualifying CompCode */

```

## MQDISC-Desconectar gerenciador de filas

A chamada MQDISC quebra a conexão entre o gerenciador de fila e o programa de aplicativo e é o inverso da chamada MQCONN ou MQCONNX.

- No z/OS, todos os aplicativos que usam consumo de mensagem assíncrona, manipulação de eventos ou retorno de chamada, o encadeamento de controle principal deve emitir uma chamada MQDISC antes de terminar. Consulte [Consumo Assíncrono de IBM MQ mensagens](#) para obter mais detalhes
- No z/OS, os aplicativos CICS não precisam emitir essa chamada para desconectar do gerenciador de filas.

Se um aplicativo CICS fizer essa chamada, ele não terá efeito, a menos que uma chamada MQCONNX anterior tenha sido feita, especificando um dos:

```

MQCNO_SERIALIZE_CONN_TAG_Q_MGR
MQCNO_SERIALIZE_CONN_TAG_QSG
MQCNO_RESTRICT_CONN_TAG_Q_MGR ou
MQCNO_RESTRICT_CONN_TAG_QSG

```

, nesse caso, todos os identificadores de objetos abertos atualmente são fechados

## Sintaxe

MQDISC (*Hconn*, *CompCode*, *Reason*)

## Parâmetros

### Hconn

Tipo: MQHCONN-entrada/saída

Esta manipulação representa a conexão ao gerenciador de filas. O valor de *Hconn* foi retornado por uma chamada anterior MQCONN ou MQCONNX.

Em z/OS para aplicativos CICS, é possível omitir a chamada MQCONN e especificar o valor a seguir para *Hconn*:::

#### MQHC\_DEF\_HCONN

Manipulação de conexões padrão.

Na conclusão bem-sucedida da chamada, o gerenciador de filas configura *Hconn* para um valor que não é um identificador válido do ambiente. Esse valor é:

#### MQHC\_UNUSABLE\_HCONN

Identificador de conexão não utilizável

No z/OS, *Hconn* é configurado para um valor indefinido.

### CompCode

Tipo: MQLONG - saída

O código de conclusão; é um dos seguintes códigos:

#### MQCC\_OK

Indica conclusão bem-sucedida.

#### MQCC\_WARNING

Aviso (conclusão parcial).

**MQCC\_FAILED**

A chamada falhou.

**Razão**

Tipo: MQLONG - saída

Se *CompCode* for MQCC\_OK:

**MQRC\_NONE**

(0, X'000') Nenhuma razão a relatar.

Se *CompCode* for MQCC\_WARNING:

**MQRC\_BACKED\_OUT**

(2003, X'7D3') Unidade de trabalho retornada.

**MQRC\_CONN\_TAG\_NOT\_LIBERADO**

(2344, X'928 ') Tag de conexão não liberada.

**MQRC\_OUTCOME\_PENDING**

(2124, X'84C') Resultado da operação de confirmação pendente.

Se *CompCode* for MQCC\_FAILED:

**MQRC\_ADAPTER\_DISC\_LOAD\_ERROR**

(2138, X'85A') Impossível carregar o módulo de desconexão do adaptador.

**MQRC\_ADAPTER\_NOT\_AVAILABLE**

(2204, X'89C') Adaptador não disponível.

**MQRC\_ADAPTER\_SERV\_LOAD\_ERROR**

(2130, X'852') Não foi possível carregar o módulo de serviço do adaptador.

**MQRC\_API\_EXIT\_ERROR**

(2374, X'946') Falha na saída de API.

**MQRC\_API\_EXIT\_INIT\_ERROR**

(2375, X'947') Falha na inicialização da saída de API.

**MQRC\_API\_EXIT\_TERM\_ERROR**

(2376, X'948') Falha na finalização da saída de API.

**MQRC\_ASID\_MISMATCH**

(2157, X'86D') ASIDs de início e primárias são diferentes.

**MQRC\_CALL\_IN\_PROGRESS**

(2219, X'8AB') Chamada MQI inserida antes da chamada anterior concluída.

**MQRC\_CONNECTION\_BROKEN**

(2009, X'7D9') Conexão ao gerenciador de filas perdida.

**MQRC\_CONNECTION\_STOPPING**

(2203, X'89B') Conexão sendo encerrada.

**MQRC\_HCONN\_ERROR**

(2018, X'7E2') Manipulação de conexões não válida.

**MQRC\_OUTCOME\_MIXED**

(2123, X'84B') O resultado da operação de confirmação ou de retorno é combinado.

**MQRC\_PAGESET\_ERROR**

(2193, X'891') Erro ao acessar o conjunto de dados do conjunto de páginas.

**MQRC\_Q\_MGR\_NAME\_ERROR**

(2058, X'80A') Nome do gerenciador de filas não válido ou não conhecido.

**MQRC\_Q\_MGR\_NOT\_AVAILABLE**

(2059, X'80B') Gerenciador de filas não disponível para conexão.

**MQRC\_Q\_MGR\_STOPPING**

(2162, X'872') Gerenciador de filas sendo encerrado.

**MQRC\_RESOURCE\_PROBLEM**

(2102, X'836') Recursos insuficientes do sistema disponíveis.

**MQRC\_STORAGE\_NOT\_AVAILABLE**

(2071, X'817') Armazenamento insuficiente disponível.

**MQRC\_UNEXPECTED\_ERROR**

(2195, X'893') Ocorreu um erro inesperado.

Para obter informações detalhadas sobre esses códigos, consulte [Mensagens e códigos de razão](#).

**Observações de Uso**

1. Se uma chamada MQDISC for emitida quando a conexão ainda tiver objetos abertos nessa conexão, o gerenciador de filas fechará esses objetos, com as opções de fechamento configuradas para MQCO\_NONE.
2. Se o aplicativo terminar com mudanças não confirmadas em uma unidade de trabalho, a disposição dessas mudanças dependerá de como o aplicativo termina:
  - a. Se o aplicativo emitir a chamada MQDISC antes de terminar:
    - Para uma unidade de trabalho coordenada pelo gerenciador de filas, o gerenciador de fila emite a chamada MQCMIT em nome do aplicativo.. A unidade de trabalho é confirmada, se possível, e restaurada, se não.
    - Para uma unidade de trabalho coordenada externamente, não há mudança no status da unidade de serviço; no entanto, o gerenciador de filas geralmente indica que a unidade de trabalho deve ser confirmada quando solicitado pelo coordenador da unidade de trabalho.  
  
Em z/OS, CICS, IMS (diferente de programas DL/1 em lote) e aplicativos RRS são assim.
  - b. Se o aplicativo terminar normalmente, mas sem emitir a chamada MQDISC, a ação tomada dependerá do ambiente:
    - No z/OS, exceto para aplicativos MQ Java ou MQ JMS , as ações descritas na nota 2a ocorrem.
    - Em todos os outros casos, as ações descritas na nota 2c ocorrem..Devido às diferenças entre os ambientes, certifique-se de que os aplicativos que você deseja portar confirmem ou recuperem a unidade de trabalho antes de terminarem.
  - c. Se o aplicativo terminar *de forma anormal* sem a emissão da chamada MQDISC, a unidade de trabalho será restaurada para fora.
3. Em z/OS, aplicam-se os seguintes pontos:
  - Os aplicativos CICS não precisam emitir a chamada MQDISC para desconectar do gerenciador de filas, porque o próprio sistema CICS se conecta ao gerenciador de filas e a chamada MQDISC não tem efeito nessa conexão.
  - CICS, IMS (diferente de programas DL/1 em lote) e aplicativos RRS usam unidades de trabalho que são coordenadas por um coordenador de unidade de trabalho externo. Como resultado, a chamada MQDISC não afeta o status da unidade de trabalho (se houver) que existe quando a chamada é emitida.  
  
No entanto, a chamada MQDISC *indica* o término de uso da tag de conexão *ConnTag* que foi associada à conexão por uma chamada MQCONNX anterior emitida pelo aplicativo. Se houver uma unidade de trabalho ativa que referencie a tag de conexão quando a chamada MQDISC for emitida, a chamada será concluída com o código de conclusão MQCC\_WARNING e o código de razão MQRC\_CONN\_TAG\_NOT\_LIBERADO A tag de conexão não se torna disponível para reutilização até que o coordenador da unidade de trabalho externo tenha resolvido a unidade de trabalho

**Nota:** No CICS, MQOP\_START não é suportado.. Em vez disso, use a chamada de função MQOP\_START\_WAIT

## Chamada C

```
MQDISC (&Hconn, &CompCode, &Reason);
```

Declare os parâmetros conforme a seguir:

```
MQHCONN  Hconn;      /* Connection handle */
MQLONG   CompCode;  /* Completion code */
MQLONG   Reason;    /* Reason code qualifying CompCode */
```

## Chamada COBOL

```
CALL 'MQDISC' USING HCONN, COMPCODE, REASON.
```

Declare os parâmetros conforme a seguir:

```
** Connection handle
01 HCONN      PIC S9(9) BINARY.
** Completion code
01 COMPCODE   PIC S9(9) BINARY.
** Reason code qualifying COMPCODE
01 REASON     PIC S9(9) BINARY.
```

## Chamada PL/I

```
call MQDISC (Hconn, CompCode, Reason);
```

Declare os parâmetros conforme a seguir:

```
dcl Hconn      fixed bin(31); /* Connection handle */
dcl CompCode   fixed bin(31); /* Completion code */
dcl Reason     fixed bin(31); /* Reason code qualifying CompCode */
```

## Chamada do assembler System/390

```
CALL MQDISC,(HCONN,COMPCODE,REASON)
```

Declare os parâmetros conforme a seguir:

```
HCONN      DS F Connection handle
COMPCODE   DS F Completion code
REASON     DS F Reason code qualifying COMPCODE
```

## Chamada do Visual Basic

```
MQDISC Hconn, CompCode, Reason
```

Declare os parâmetros conforme a seguir:

```
Dim Hconn      As Long 'Connection handle'
Dim CompCode   As Long 'Completion code'
Dim Reason     As Long 'Reason code qualifying CompCode'
```

## MQDLTMH-Excluir identificador de mensagem

A chamada MQDLTMH exclui um identificador de mensagens e é o inverso da chamada MQCRTMH

### Sintaxe

MQDLTMH (*Hconn, Hmsg, DltMsgHOpts, CompCode, Reason*)

### Parâmetros

#### Hconn

Tipo: MQHCONN - entrada

Esta manipulação representa a conexão ao gerenciador de filas.

O valor deve corresponder ao identificador de conexão que foi usado para criar o identificador de mensagem especificado no parâmetro **Hmsg ...**

Se a manipulação de mensagens foi criada usando MQHC\_UNASSOCIATED\_HCONN, uma conexão válida deverá ser estabelecida no encadeamento excluindo a manipulação de mensagens, caso contrário, a chamada falhará com MQRC\_CONNECTION\_BROKEN

#### Hmsg

Tipo: MQHMSG-entrada/saída

Esta é a manipulação de mensagem a ser excluída O valor foi retornado por uma chamada MQCRTMH anterior.

Na conclusão bem-sucedida da chamada, o identificador é configurado para um valor inválido para o ambiente. Esse valor é:.

#### **MQHM\_UNUSABLE\_HMSG**

Identificador de mensagem não utilizável

O identificador de mensagem não poderá ser excluído se outra chamada IBM MQ estiver em andamento e tiver passado o mesmo identificador de mensagem...

#### DltMsgHOpts

Tipo: MQDMHO-entrada

Consulte [MQDMHO](#) para obter detalhes..

#### CompCode

Tipo: MQLONG - saída

O código de conclusão; é um dos seguintes:

#### **MQCC\_OK**

Indica conclusão bem-sucedida.

#### **MQCC\_FAILED**

A chamada falhou.

#### Razão

Tipo: MQLONG - saída

Se *CompCode* for MQCC\_OK:

#### **MQRC\_NONE**

(0, X'000') Nenhuma razão a relatar.

Se *CompCode* for MQCC\_FAILED:

#### **MQRC\_ADAPTER\_NOT\_AVAILABLE**

(2204, X'089C') Adaptador não disponível.

#### **MQRC\_ADAPTER\_SERV\_LOAD\_ERROR**

(2130, X'852') Não foi possível carregar o módulo de serviço do adaptador.

**MQRC\_ASID\_MISMATCH**

(2157, X'86D') ASIDs de início e primárias são diferentes.

**MQRC\_CALL\_IN\_PROGRESS**

(2219, X'08AB') chamada MQI inserida antes da chamada anterior ser concluída.

**MQRC\_CONNECTION\_BROKEN**

(2009, X'07D9') Conexão com o gerenciador de filas perdida.

**MQRC\_DMHO\_ERROR**

(2462, X'099E') A estrutura de opções de manipulação de mensagens de exclusão não é válida...

**MQRC\_HMSG\_ERROR**

(2460, X'099C') Ponteiro do identificador de mensagens inválido.

**MQRC\_MSG\_HANDLE\_IN\_USE**

(2499, X'09C3') Identificador de mensagem já em uso.

**MQRC\_OPTIONS\_ERROR**

(2046, X'07FE') Opções não válidas ou não consistentes.

**MQRC\_STORAGE\_NOT\_AVAILABLE**

(2071, X'817') Armazenamento insuficiente disponível.

**MQRC\_UNEXPECTED\_ERROR**

(2195, X'893') Ocorreu um erro inesperado.

Para obter informações detalhadas sobre esses códigos, consulte [Mensagens e códigos de razão](#).

## Chamada C

```
MQDLTMH (Hconn, &Hmsg, &DltMsgHOpts, &CompCode, &Reason);
```

Declare os parâmetros conforme a seguir:

```
MQHCONN  Hconn;          /* Connection handle */
MQHMSG   Hmsg;          /* Message handle */
MQDMHO   DltMsgHOpts;  /* Options that control the action of MQDLTMH */
MQLONG   CompCode;     /* Completion code */
MQLONG   Reason;       /* Reason code qualifying CompCode */
```

## Chamada COBOL

```
CALL 'MQDLTMH' USING HCONN, HMSG, DLTMSGHOPTS, COMPCODE, REASON.
```

Declare os parâmetros conforme a seguir:

```
** Connection handle
01  HCONN    PIC S9(9) BINARY.

** Options that control the action of MQDLTMH
01  DLTMSGHOPTS.
   COPY CMQDMHOL.

** Completion code
01  COMPCODE  PIC S9(9) BINARY.

** Reason code qualifying COMPCODE
01  REASON    PIC S9(9) BINARY.
```

## Chamada PL/I

```
call MQDLTMH (Hconn, Hmsg, DltMsgHOpts, CompCode, Reason);
```

Declare os parâmetros conforme a seguir:

```
dcl Hconn          /* Connection handle */
dcl Hmsg          /* Message handle */
dcl DltMsgHOpts  like MQDMHO; /* Options that control the action of MQDLTMH */
dcl CompCode     /* Completion code */
dcl Reason       /* Reason code qualifying CompCode */
```

## Chamada do Assembler de Alto Nível

```
CALL MQDLTMH, (HCONN, HMSG, DLTMMSGHOPTS, COMPCODE, REASON)
```

Declare os parâmetros conforme a seguir:

HCONN	DS	F	Connection handle
HMSG	DS	D	Message handle
DLTMMSGHOPTS	CMQDMHOA	,	Options that control the action of MQDLTMH
COMPCODE	DS	F	Completion code
REASON	DS	F	Reason code qualifying COMPCODE

## MQDLTMP-Excluir propriedade de mensagem

A chamada MQDLTMP exclui uma propriedade de um identificador de mensagem e é o inverso da chamada MQSETMP

### Sintaxe

MQDLTMP (*Hconn, Hmsg, DltPropOpts, Nome, CompCode, Razão*)

### Parâmetros

#### Hconn

Tipo: MQHCONN - entrada

Esta manipulação representa a conexão ao gerenciador de filas. O valor deve corresponder ao identificador de conexão que foi usado para criar o identificador de mensagem especificado no parâmetro **Hmsg** ...

Se a manipulação de mensagens foi criada usando MQHC\_UNASSOCIATED\_HCONN, uma conexão válida deverá ser estabelecida no encadeamento excluindo a manipulação de mensagens, caso contrário, a chamada falhará com MQRC\_CONNECTION\_BROKEN

#### Hmsg

Tipo: MQHMSG-entrada

Esta é a manipulação de mensagem que contém a propriedade a ser excluída. O valor foi retornado por uma chamada MQCRTMH anterior.

#### DltPropOpts

Tipo: MQDMPO-entrada

Consulte o tipo de dados [MQDMPO](#) para obter detalhes..

#### Nome

Tipo: MQCHARV-entrada

O nome da propriedade a excluir. Consulte [Nomes de Propriedades](#) , para obter informações adicionais sobre nomes de propriedade

Os curingas não são permitidos no nome da propriedade

#### CompCode

Tipo: MQLONG - saída

O código de conclusão; é um dos seguintes:

**MQCC\_OK**

Indica conclusão bem-sucedida.

**MQCC\_WARNING**

Aviso (conclusão parcial).

**MQCC\_FAILED**

A chamada falhou.

**Razão**

Tipo: MQLONG - saída

Se *CompCode* for MQCC\_OK:

**MQRC\_NONE**

(0, X'000') Nenhuma razão a relatar.

Se *CompCode* for MQCC\_WARNING:

**MQRC\_PROPERTY\_NOT\_AVAILABLE**

(2471, X'09A7') Propriedade não disponível.

**MQRC\_RFH\_FORMAT\_ERROR**

(2421, X'0975 ') Uma pasta MQRFH2 contendo propriedades não pôde ser analisada.

Se *CompCode* for MQCC\_FAILED:

**MQRC\_ADAPTER\_NOT\_AVAILABLE**

(2204, X'089C') Adaptador não disponível.

**MQRC\_ADAPTER\_SERV\_LOAD\_ERROR**

(2130, X'0852 ') Impossível carregar o módulo de serviço do adaptador.

**MQRC\_ASID\_MISMATCH**

(2157, X'086D') Os ASIDs primário e inicial diferem.

**MQRC\_CALL\_IN\_PROGRESS**

(2219, X'08AB') chamada MQI inserida antes da chamada anterior ser concluída.

**MQRC\_CONNECTION\_BROKEN**

(2009, X'07D9') Conexão com o gerenciador de filas perdida.

**MQRC\_DMPO\_ERROR**

(2481, X'09B1') A estrutura de opções de propriedade de mensagem de exclusão não é válida..

**MQRC\_HMSG\_ERROR**

(2460, X'099C') Identificador de mensagem inválido.

**MQRC\_MSG\_HANDLE\_IN\_USE**

(2499, X'09C3') Identificador de mensagem já em uso.

**MQRC\_OPTIONS\_ERROR**

(2046, X'07FE') Opções não válidas ou não consistentes.

**MQRC\_PROPERTY\_NAME\_ERROR**

(2442, X'098A') Nome da propriedade inválido

**MQRC\_SOURCE\_CCSID\_ERROR**

(2111, X'083F') Identificador do conjunto de caracteres codificados do nome da propriedade inválido.

**MQRC\_UNEXPECTED\_ERROR**

(2195, X'0893 ') Ocorreu um erro inesperado.

Para obter informações detalhadas sobre esses códigos, consulte [Mensagens e códigos de razão](#).



## Chamada C

```
MQDLTMP (Hconn, Hmsg, &DltPropOpts, &Name, &CompCode, &Reason)
```

Declare os parâmetros conforme a seguir:

```
MQHCONN Hconn;      /* Connection handle */
MQHMSG  Hmsg;       /* Message handle */
MQDMPO  DltPropOpts; /* Options that control the action of MQDLTMP */
MQCHARV Name;       /* Property name */
MQLONG  CompCode;   /* Completion code */
MQLONG  Reason;     /* Reason code qualifying CompCode */
```

## Chamada COBOL

```
CALL 'MQDLTMP' USING HCONN, HMSG, DLTPROPOPTS, NAME, COMPCODE, REASON.
```

Declare os parâmetros conforme a seguir:

```
** Connection handle
01 HCONN    PIC S9(9) BINARY.
** Message handle
01 HMSG     PIC S9(18) BINARY.
** Options that control the action of MQDLTMP
01 DLTPROPOPTS.
   COPY CMQDMPOV.
** Property name
01 NAME.
   COPY CMQCHRVA.
** Completion code
01 COMPCODE PIC S9(9) BINARY.
** Reason code qualifying COMPCODE
01 REASON  PIC S9(9) BINARY.
```

## Chamada PL/I

```
call MQDLTMP (Hconn, Hmsg, DltPropOpts, Name, CompCode, Reason);
```

Declare os parâmetros conforme a seguir:

```
dcl Hconn      fixed bin(31); /* Connection handle */
dcl Hmsg       fixed bin(63); /* Message handle */
dcl DltPropOpts like MQDMPO;  /* Options that control the action of MQDLTMP */
dcl Name       like MQCHARV;  /* Property name */
dcl CompCode   fixed bin(31); /* Completion code */
dcl Reason     fixed bin(31); /* Reason code qualifying CompCode */
```

## Chamada do Assembler de Alto Nível

```
CALL MQDLTMP, (HCONN, HMSG, DLTPROPOPTS, NAME, COMPCODE, REASON)
```

Declare os parâmetros conforme a seguir:

```
HCONN      DS      F      Connection handle
HMSG       DS      D      Message handle
DLTPROPOPTS CMQDMPOA ,      Options that control the action of MQDLTMP
NAME       CMQCHRVA ,      Property name
COMPCODE   DS      F      Completion code
REASON     DS      F      Reason code qualifying COMPCODE
```

## MQGET - Obter Mensagem

A chamada MQGET recupera uma mensagem de uma fila local que foi aberta utilizando a chamada MQOPEN.

### Sintaxe

MQGET (*Hconn*, *Hobj*, *MsgDesc*, *GetMsgOpts*, *BufferLength*, *Buffer*, *DataLength*, *CompCode*, *Reason*)

### Parâmetros

#### Hconn

Tipo: MQHCONN - entrada

Esta manipulação representa a conexão ao gerenciador de filas. O valor de *Hconn* foi retornado por uma chamada anterior MQCONN ou MQCONNX.

Em z/OS para aplicativos CICS, a chamada MQCONN pode ser omitida e o seguinte valor especificado para *Hconn*:

#### MQHC\_DEF\_HCONN

Manipulação de conexões padrão.

#### Hobj

Tipo: MQHOBJ - entrada

Esse manipulador representa a fila da qual uma mensagem será recuperada. O valor de *Hobj* foi retornado por uma chamada MQOPEN anterior. A fila deve ter sido aberta com uma ou mais das seguintes opções (consulte [“MQOPEN-Abrir objeto”](#) na página 759 para detalhes):

- MQOO\_INPUT\_SHARED
- MQOO\_INPUT\_EXCLUSIVE
- MQOO\_INPUT\_AS\_Q\_DEF
- MQOO\_BROWSE

#### MsgDesc

Tipo: MQMD - entrada/saída

Essa estrutura descreve os atributos da mensagem requerida e os atributos da mensagem recuperada. Consulte [“MQMD - descritor de mensagem”](#) na página 435 para obter detalhes.

Se *BufferLength* for menor que o comprimento da mensagem, *MsgDesc* será preenchido pelo gerenciador de fila, se MQGMO\_ACCEPT\_TRUNCATED\_MSG for especificado no parâmetro **GetMsgOpts** (consulte [MQGMO-Options field](#)).

Se o aplicativo fornecer um MQMD version-1, a mensagem retornada terá um MQMDE prefixado para os dados da mensagem do aplicativo, mas somente se um ou mais campos no MQMDE tiverem um valor não padrão. Se todos os campos no MQMDE tiver valores padrão, o MQMDE será omitido. Um nome do formato de MQFMT\_MD\_EXTENSION no campo *Formato* em MQMD indica que um MQMDE está presente.

O aplicativo não precisa fornecer uma estrutura de MQMD se uma manipulação de mensagem válida é fornecida no campo *MsgHandle*. Se nada for fornecido neste campo, o descritor da mensagem é obtido do descritor associado com as manipulações de mensagem.

Se o aplicativo fornecer uma manipulação de mensagem em vez de uma estrutura de MQMD e especificar MQGMO\_PROPERTIES\_FORCE\_MQRFH2, a chamada falhará com código de razão MQRC\_MD\_ERROR. A chamada também falha, com o código de razão MQRC\_MD\_ERROR, se o aplicativo não fornecer uma estrutura MQMD e especificar MQGMO\_PROPERTIES\_AS\_Q\_DEF e o atributo da fila **PropertyControl** for MQPROP\_FORCE\_MQRFH2.

Se opções correspondentes forem especificadas e o descritor de mensagens associado com a manipulação de mensagem estiver sendo utilizado, os campos de entrada utilizados para correspondência serão provenientes da manipulação de mensagem.

## GetMsgOpts

Tipo: MQGMO - entrada/saída

Consulte “MQGMO-Obter opções de mensagem” na página 379 para obter detalhes.

## BufferLength

Tipo: MQLONG - entrada

Este é o comprimento em bytes da área *Buffer*. Especifique zero para mensagens que não possuem dados ou se a mensagem será removida da fila e os dados descartados (você deve especificar MQGMO\_ACCEPT\_TRUNCATED\_MSG neste caso).

**Nota:** O comprimento da mensagem mais longa que é possível ler na fila é fornecido pelo atributo da fila **MaxMsgLength**; consulte “Atributos para filas” na página 865.

## de Saída

Tipo: MQBYTEExBufferLength - saída

Este é a área para conter os dados da mensagem de dados. Alinhe o buffer em um limite apropriado para a natureza dos dados na mensagem. O alinhamento de 4 bytes é adequado para a maioria das mensagens (incluindo mensagens contendo estruturas de cabeçalho IBM MQ), mas algumas mensagens podem requerer alinhamento mais rigoroso. Por exemplo, uma mensagem contendo um número inteiro binário de 64 bits pode requerer alinhamento de 8 bytes.

Se *BufferLength* for menor que o comprimento da mensagem, o máximo possível da mensagem será movido para **Buffer**. Isso acontece se MQGMO\_ACCEPT\_TRUNCATED\_MSG for especificado no parâmetro **GetMsgOpts** (consulte campo MQGMO-Options para obter mais informações).

O conjunto de caracteres e a codificação dos dados em **Buffer** são fornecidos pelos campos *CodedCharSetId* e *Encoding* retornados no parâmetro **MsgDesc**. Se esses valores forem diferentes dos valores requeridos pelo receptor, o receptor deverá converter os dados da mensagem do aplicativo para o conjunto de caracteres e codificação requeridos. A opção MQGMO\_CONVERT pode ser usada (com uma saída gravada pelo usuário, se necessário) para converter os dados da mensagem; consulte “MQGMO-Obter opções de mensagem” na página 379 para detalhes dessa opção.

**Nota:** Todos os outros parâmetros na chamada MQGET estão no conjunto de caracteres e na codificação do gerenciador de fila local (fornecido pelo atributo do gerenciador de filas **CodedCharSetId** e MQENC\_NATIVE).

Se a chamada falhar, o conteúdo do buffer pode ainda ter mudado.

Na linguagem de programação C, o parâmetro é declarado como um ponteiro a evitar: o endereço de qualquer tipo de dados pode ser especificado como o parâmetro.

Se o parâmetro **BufferLength** for zero, *Buffer* não será referido; nesse caso, o endereço do parâmetro transmitido por programas gravados em C ou no assembler System/390 poderá ser nulo...

## DataLength

Tipo: MQLONG - saída

Este é o comprimento em bytes dos dados do aplicativo *na mensagem*. Se esse valor for maior que *BufferLength*, somente *BufferLength* bytes serão retornados no parâmetro **Buffer** (ou seja, a mensagem será truncada).. Se o valor for zero, a mensagem não conterá nenhum dado do aplicativo.

Se *BufferLength* for menor que o comprimento da mensagem, *DataLength* ainda será concluído pelo gerenciador de filas, se MQGMO\_ACCEPT\_TRUNCATED\_MSG for especificado no parâmetro **GetMsgOpts** (consulte campo MQGMO-Options para obter mais informações). Isso permite que o aplicativo determine o tamanho do buffer necessário para acomodar os dados da mensagem, e depois emita novamente a chamada com um buffer do tamanho apropriado.

No entanto, se a opção MQGMO\_CONVERT for especificada e os dados da mensagem convertidos forem muito longos para caber em *Buffer*, o valor retornado para *DataLength* será:

- O comprimento dos dados *não convertidos* para formatos definidos pelo gerenciador de filas.

Nesse caso, se a natureza dos dados fizer com que sejam expandidos durante a conversão, o aplicativo deverá alocar um buffer maior que o valor retornado pelo gerenciador de filas para *DataLength*.

- O valor retornado pela saída de conversão de dados, para formatos definidos pelo aplicativo.

### CompCode

Tipo: MQLONG - saída

O código de conclusão; é um dos seguintes:

#### **MQCC\_OK**

Indica conclusão bem-sucedida.

#### **MQCC\_WARNING**

Aviso (conclusão parcial).

#### **MQCC\_FAILED**

A chamada falhou.

### Razão

Tipo: MQLONG - saída

Os códigos de razão listados são aqueles que o Gerenciador de Filas pode retornar para o parâmetro **Reason** Se o aplicativo especificar a opção MQGMO\_CONVERT e uma saída gravada pelo usuário for chamada para converter alguns ou todos os dados da mensagem, a saída decidirá qual valor será retornado para o parâmetro **Reason** . Como resultado outros valores diferentes dos valores documentos são possíveis.

Se *CompCode* for MQCC\_OK:

#### **MQRC\_NONE**

(0, X'000') Nenhuma razão a relatar.

Se *CompCode* for MQCC\_WARNING:

#### **MQRC\_CONVERTED\_MSG\_TOO\_BIG**

(2120, X'848') Dados convertidos muito grandes para o buffer.

#### **MQRC\_CONVERTED\_STRING\_TOO\_BIG**

(2190, X'88E') Sequência convertida muito grande para o campo.

#### **MQRC\_DBCS\_ERROR**

(2150, X'866') Sequência DBCS inválida.

#### **MQRC\_FORMAT\_ERROR**

(2110, X'83E') Formato da mensagem inválido.

#### **MQRC\_INCOMPLETE\_GROUP**

(2241, X'8C1') Grupo de mensagens não concluído.

#### **MQRC\_INCOMPLETE\_MSG**

(2242, X'8C2') Mensagem lógica não concluída.

#### **MQRC\_INCONSISTENT\_CCSIDS**

(2243, X'8C3') Segmentos da mensagem possuem CCSIDs diferentes.

#### **MQRC\_INCONSISTENT\_ENCODINGS**

(2244, X'8C4') Segmentos da mensagem possuem codificações diferentes.

#### **MQRC\_INCONSISTENT\_UOW**

(2245, X'8C5') Especificação de unidade de trabalho inconsistente.

#### **MQRC\_MSG\_TOKEN\_ERROR**

(2331, X'91B') Uso inválido de token da mensagem.

#### **MQRC\_NO\_MSG\_LOCKED**

(2209, X'8A1') Nenhuma mensagem bloqueada.

#### **MQRC\_NOT\_CONVERTED**

(2119, X'847') Dados da mensagem não convertidos.

**MQRC\_OPTIONS\_CHANGED**

(nnnn, X'xxx') Opções que deveriam ser consistentes foram mudadas.

**MQRC\_PARTIALLY\_CONVERTED**

(2272, X'8E0') Dados da mensagem parcialmente convertidos.

**MQRC\_SIGNAL\_REQUEST\_ACCEPTED**

(2070, X'816') Nenhuma mensagem retornada (mas solicitação de sinal aceita).

**MQRC\_SOURCE\_BUFFER\_ERROR**

(2145, X'861') Parâmetro de buffer de origem inválido.

**MQRC\_SOURCE\_CCSID\_ERROR**

(2111, X'83F') Identificador do conjunto de caracteres codificados da origem inválido.

**MQRC\_SOURCE\_DECIMAL\_ENC\_ERROR**

(2113, X'841') Codificação decimal compactada na mensagem não reconhecida.

**MQRC\_SOURCE\_FLOAT\_ENC\_ERROR**

(2114, X'842') Codificação de ponto flutuante na mensagem não reconhecida.

**MQRC\_SOURCE\_INTEGER\_ENC\_ERROR**

(2112, X'840') Codificação de número inteiro da origem não reconhecida.

**MQRC\_SOURCE\_LENGTH\_ERROR**

(2143, X'85F') Parâmetro de comprimento de origem inválido.

**MQRC\_TARGET\_BUFFER\_ERROR**

(2146, X'862') Parâmetro de buffer de destino inválido.

**MQRC\_TARGET\_CCSID\_ERROR**

(2115, X'843') Identificador do conjunto de caracteres codificados do destino inválido.

**MQRC\_TARGET\_DECIMAL\_ENC\_ERROR**

(2117, X'845') Codificação decimal compactada especificada pelo receptor não reconhecida.

**MQRC\_TARGET\_FLOAT\_ENC\_ERROR**

(2118, X'846') Codificação de ponto flutuante especificada pelo receptor não reconhecida.

**MQRC\_TARGET\_INTEGER\_ENC\_ERROR**

(2116, X'844') Codificação de número inteiro do destino não reconhecida.

**MQRC\_TRUNCATED\_MSG\_ACCEPTED**

(2079, X'81F') Mensagem truncada retornada (processamento concluído).

**MQRC\_TRUNCATED\_MSG\_FAILED**

(2080, X'820') Mensagem truncada retornada (processamento não concluído).

Se *CompCode* for MQCC\_FAILED:

**MQRC\_ADAPTER\_NOT\_AVAILABLE**

(2204, X'89C') Adaptador não disponível.

**MQRC\_ADAPTER\_CONV\_LOAD\_ERROR**

(2133, X'855') Impossível carregar módulos de serviços de conversão de dados.

**MQRC\_ADAPTER\_SERV\_LOAD\_ERROR**

(2130, X'852') Não foi possível carregar o módulo de serviço do adaptador.

**MQRC\_API\_EXIT\_ERROR**

(2374, X'946') Falha na saída de API.

**MQRC\_API\_EXIT\_LOAD\_ERROR**

(2183, X'887') Não foi possível carregar a saída de API.

**MQRC\_ASID\_MISMATCH**

(2157, X'86D') ASIDs de início e primárias são diferentes.

**MQRC\_BACKED\_OUT**

(2003, X'7D3') Unidade de trabalho retornada.

**MQRC\_BUFFER\_ERROR**

(2004, X'7D4') Parâmetro de buffer inválido.

**MQRC\_BUFFER\_LENGTH\_ERROR**

(2005, X'7D5') Parâmetro de comprimento de buffer inválido.

**MQRC\_CALL\_IN\_PROGRESS**

(2219, X'8AB') Chamada MQI inserida antes da chamada anterior concluída.

**MQRC\_CF\_NOT\_AVAILABLE**

(2345, X' 929 ') Recurso de acoplamento não disponível.

**MQRC\_CF\_STRUC\_FAILED**

(2373, X'945') Falha na estrutura do recurso de acoplamento.

**MQRC\_CF\_STRUC\_IN\_USE**

(2346, X'92A') Estrutura de recurso de acoplamento em uso.

**MQRC\_CF\_STRUC\_LIST\_HDR\_IN\_USE**

(2347, X'92B') Cabeçalho da lista da estrutura do recurso de acoplamento em uso.

**MQRC\_CICS\_WAIT\_FAILED**

(2140, X'85C') Pedido de espera rejeitado por CICS.

**MQRC\_CONNECTION\_BROKEN**

(2009, X'7D9') Conexão ao gerenciador de filas perdida.

**MQRC\_CONNECTION\_NOT\_AUTHORIZED**

(2217, X'8A9') Não autorizado para conexão.

**MQRC\_CONNECTION QUIESCING**

(2202, X'89A') Quiesce de conexão.

**MQRC\_CONNECTION\_STOPPING**

(2203, X'89B') Conexão sendo encerrada.

**MQRC\_CORREL\_ID\_ERROR**

(2207, X'89F') Erro de identificador de correlação.

**MQRC\_DATA\_LENGTH\_ERROR**

(2010, X'7DA') Parâmetro de comprimento de dados inválido.

**MQRC\_DB2\_NOT\_AVAILABLE**

(2342, X' 926 ') Db2 subsistema não disponível.

**MQRC\_GET\_INHIBITED**

(2016, X'7E0 ') É inibida para a fila.

**MQRC\_GLOBAL\_UOW\_CONFLICT**

(2351, X'92F') Unidades de trabalho globais conflitam.

**MQRC\_GMO\_ERROR**

(2186, X'88A') Estrutura de opções de obtenção de mensagem inválida.

**MQRC\_HANDLE\_IN\_USE\_FOR\_UOW**

(2353, X'931') Manipulação em uso para unidade de trabalho global.

**MQRC\_HCONN\_ERROR**

(2018, X'7E2') Manipulação de conexões não válida.

**MQRC\_HOBJ\_ERROR**

(2019, X'7E3') Manipulação de objetos não válida.

**MQRC\_INCONSISTENT\_BROWSE**

(2259, X'8D3') Especificação de pesquisa inconsistente.

**MQRC\_INCONSISTENT\_UOW**

(2245, X'8C5') Especificação de unidade de trabalho inconsistente.

**MQRC\_INVALID\_MSG\_UNDER\_CURSOR**

(2246, X'8C6 ') Mensagem sob o cursor inválida para recuperação.

**MQRC\_LOCAL\_UOW\_CONFLICT**

(2352, X'930') Unidade de trabalho global conflita com unidade de trabalho local.

**MQRC\_MATCH\_OPTIONS\_ERROR**

(2247, X'8C7') Opções correspondentes inválidas.

**MQRC\_MD\_ERROR**  
(2026, X'7EA') Descritores de mensagem inválidos.

**MQRC\_MSG\_ID\_ERROR**  
(2206, X'89E') Erro de identificador de mensagem.

**MQRC\_MSG\_SEQ\_NUMBER\_ERROR**  
(2250, X'8CA') Número de sequência da mensagem inválido.

**MQRC\_MSG\_TOKEN\_ERROR**  
(2331, X'91B') Uso de token da mensagem inválido.

**MQRC\_NO\_MSG\_AVAILABLE**  
(2033, X'7F1') Nenhuma mensagem disponível.

**MQRC\_NO\_MSG\_UNDER\_CURSOR**  
(2034, X'7F2') Cursor de navegação não posicionado na mensagem.

**MQRC\_NOT\_OPEN\_FOR\_BROWSE**  
(2036, X'7F4') Fila não aberta para navegação.

**MQRC\_NOT\_OPEN\_FOR\_INPUT**  
(2037, X'7F5') Fila não aberta para entrada.

**MQRC\_OBJECT\_CHANGED**  
(2041, X'7F9') Definição de objeto alterada desde a abertura.

**MQRC\_OBJECT\_DAMAGED**  
(2101, X'835') Objeto danificado.

**MQRC\_OPTIONS\_ERROR**  
(2046, X'7FE') Opções não válidas ou não consistentes.

**MQRC\_PAGESET\_ERROR**  
(2193, X'891') Erro ao acessar o conjunto de dados do conjunto de páginas.

**MQRC\_Q\_DELETED**  
(2052, X'804') A fila foi excluída.

**MQRC\_Q\_INDEX\_TYPE\_ERROR**  
(2394, X'95A') Fila possui tipo de índice errado.

**MQRC\_Q\_MGR\_NAME\_ERROR**  
(2058, X'80A') Nome do gerenciador de filas não válido ou não conhecido.

**MQRC\_Q\_MGR\_NOT\_AVAILABLE**  
(2059, X'80B') Gerenciador de filas não disponível para conexão.

**MQRC\_Q\_MGR QUIESCING**  
(2161, X'871') Quiesce do gerenciador de filas.

**MQRC\_Q\_MGR\_STOPPING**  
(2162, X'872') Gerenciador de filas sendo encerrado.

**MQRC\_RESOURCE\_PROBLEM**  
(2102, X'836') Recursos insuficientes do sistema disponíveis.

**MQRC\_SECOND\_MARK\_NOT\_ALLOWED**  
(2062, X'80E') Uma mensagem já está marcada.

**MQRC\_SIGNAL\_OUTSTANDING**  
(2069, X'815') Sinais pendentes para essa manipulação.

**MQRC\_SIGNAL1\_ERROR**  
(2099, X'833') Campo de sinal inválido.

**MQRC\_STORAGE\_MEDIUM\_FULL**  
(2192, X'890') O meio de armazenamento externo está cheio.

**MQRC\_STORAGE\_NOT\_AVAILABLE**  
(2071, X'817') Armazenamento insuficiente disponível.

**MQRC\_SUPPRESSED\_BY\_EXIT**  
(2109, X'83D') Chamada suprimida pelo programa de saída.

**MQRC\_SYNCPOINT\_LIMIT\_REACHED**

(2024, X'7E8') Nenhuma mensagem adicional pode ser manipulada dentro da unidade de trabalho atual.

**MQRC\_SYNCPOINT\_NOT\_AVAILABLE**

(2072, X'818') Suporte de ponto de sincronização não disponível.

**MQRC\_UNEXPECTED\_ERROR**

(2195, X'893') Ocorreu um erro inesperado.

**MQRC\_UOW\_ENLISTMENT\_ERROR**

(2354, X'932') Cadastramento na unidade de trabalho global falhou.

**MQRC\_UOW\_MIX\_NOT\_SUPPORTED**

(2355, X'933') Combinação de chamadas de unidade de trabalho não suportada.

**MQRC\_UOW\_NOT\_AVAILABLE**

(2255, X'8CF') Unidade de trabalho não disponível para o gerenciador de filas a usar.

**MQRC\_WAIT\_INTERVAL\_ERROR**

(2090, X'82A') Intervalo de Espera em MQGMO inválido.

**MQRC\_WRONG\_GMO\_VERSION**

(2256, X'8D0') Versão errada do MQGMO fornecido.

**MQRC\_WRONG\_MD\_VERSION**

(2257, X'8D1') Versão errada do MQMD fornecido.

Para obter informações detalhadas sobre esses códigos, consulte [Mensagens e códigos de razão](#).

## Observações de Uso

1. A mensagem recuperada normalmente é excluída da fila. Esta exclusão pode ocorrer como parte da própria chamada MQGET ou como parte de um ponto de sincronização.

As opções de navegação são: MQGMO\_BROWSE\_FIRST, MQGMO\_BROWSE\_NEXT e MQGMO\_BROWSE\_MSG\_UNDER\_CURSOR.

2. Se a opção MQGMO\_LOCK for especificada com uma das opções de navegação, a mensagem navegada será bloqueada para que ela esteja visível somente para este identificador.

Se a opção MQGMO\_UNLOCK for especificada, uma mensagem bloqueada anteriormente será desbloqueada. Nenhuma mensagem é recuperada neste caso e os parâmetros **MsgDesc**, **BufferLength** e **BufferDataLength** não são verificados ou alterados.

3. Para aplicativos que emitem uma chamada MQGET, a mensagem recuperada pode ser perdida se o aplicativo finalizar de maneira anormal ou a conexão for grave durante o processamento da chamada. Esse problema surge porque o substituto em execução na mesma plataforma que o gerenciador de fila que emite a chamada MQGET em nome do aplicativo não pode detectar a perda do aplicativo até que o substituto esteja prestes a retornar a mensagem para o aplicativo, após a mensagem ter sido removida da fila. Esse problema pode ocorrer para ambas as mensagens, persistentes e não persistentes.

Para eliminar o risco de perder mensagens dessa maneira, sempre recupere mensagens dentro de unidades de trabalho. Isto é, especificando a opção MQGMO\_SYNCPOINT opção na chamada MQGET, e utilizando as chamadas MQCMIT ou MQBACK para confirmar ou reverter a unidade de trabalho quando o processamento de mensagem for concluído. Se MQGMO\_SYNCPOINT é especificado e o cliente é finalizado de maneira anormal ou a conexão é interrompida, o substituto restaura a unidade de trabalho no gerenciador de filas e a mensagem é recolocada na fila. Para obter mais informações sobre pontos de sincronização, consulte [Considerações de ponto de sincronização em IBM MQ aplicativos](#).

Essa situação pode surgir com clientes IBM MQ, bem como com aplicativos que estão em execução na mesma plataforma que o gerenciador de filas.



4. Se um aplicativo colocar uma sequência de mensagens em uma fila específica em uma única unidade de trabalho e, em seguida, confirmar essa unidade de trabalho com êxito, as mensagens ficarão disponíveis para recuperação conforme a seguir:

- Se a fila for uma *fila não compartilhada* (ou seja, uma fila local), todas as mensagens na unidade de trabalho se tornarão disponíveis ao mesmo tempo.
- Se a fila for uma *fila compartilhada*, as mensagens na unidade de trabalho se tornarão disponíveis na ordem em que foram colocadas, mas não todas ao mesmo tempo. Quando o sistema está excessivamente carregado, é possível que a primeira mensagem na unidade de trabalho seja recuperada com êxito, mas a chamada MQGET para a segunda mensagem ou para a mensagem subsequente na unidade de trabalho falhará com MQRC\_NO\_MSG\_AVAILABLE. Se isto ocorrer, o aplicativo deverá aguardar um curto período de tempo e, em seguida, tentar a operação novamente.

5. Se um aplicativo colocar uma sequência de mensagens na mesma fila sem usar grupos de mensagens, a ordem dessas mensagens será preservada desde que determinadas condições sejam satisfeitas. Consulte [Observações de Uso de MQPUT](#) para obter detalhes. Se as condições forem satisfeitas, as mensagens serão apresentadas no aplicativo de recebimento na ordem na qual elas foram enviadas, desde que:

- Somente um receptor esteja recebendo mensagens da fila.

Se houver dois ou mais aplicativos recebendo mensagens da fila, eles deverão concordar com o emissor que o mecanismo seja usado para identificar mensagens que pertencem a uma sequência. Por exemplo, o emissor pode configurar todos os campos `CorrelId` nas mensagens em uma sequência para um valor que era exclusivo para essa sequência de mensagens...

- O receptor não altera deliberadamente a ordem de recuperação, por exemplo, especificando um determinado `MsgId` ou `CorrelId`

Se os aplicativos de envio colocarem as mensagens como um grupo de mensagens, as mensagens serão apresentadas ao aplicativo de recebimento na ordem correta desde que o aplicativo de recebimento especifique a opção `MQGMO_LOGICAL_ORDER` na chamada MQGET. Para obter informações adicionais sobre grupos de mensagens, consulte:

- [MQMD - Campo MsgFlags](#)
- [MQPMO\\_LOGICAL\\_ORDER](#)
- [MQGMO\\_LOGICAL\\_ORDER](#)

Se o usuário estiver recebendo mensagens em um grupo no ponto de sincronização, elas deverão assegurar que o grupo completo seja processado antes de tentar concluir a transação.

6. Os aplicativos devem testar o código de feedback `MQFB_QUIT` no campo Feedback do parâmetro **MsgDesc** e terminar se eles localizarem esse valor. Consulte [MQMD - Campo Feedback](#) para obter informações adicionais.

7. Se a fila identificada por `Hobj` tiver sido aberta com a opção `MQOO_SAVE_ALL_CONTEXT` e o código de conclusão da chamada MQGET for `MQCC_OK` ou `MQCC_WARNING`, o contexto associado ao identificador de fila `Hobj` será configurado para o contexto da mensagem que foi recuperada (a menos que a opção `MQGMO_BROWSE_FIRST`, `MQGMO_BROWSE_NEXT` ou `MQGMO_BROWSE_MSG_UNDER_CURSOR` esteja configurada, nesse caso, o contexto será marcado como não disponível)..

É possível usar o contexto salvo em uma chamada MQPUT ou MQPUT1 subsequente especificando as opções `MQPMO_PASS_IDENTITY_CONTEXT` ou `MQPMO_PASS_ALL_CONTEXT`. Isto permite que o contexto da mensagem recebida seja transferido no todo ou em parte para outra mensagem (por exemplo, quando a mensagem é encaminhada para outra fila). Para obter mais informações sobre contexto da mensagem, consulte [Contexto da mensagem](#).

8. Se você incluir a opção `MQGMO_CONVERT` no parâmetro **GetMsgOpts**, os dados da mensagem do aplicativo serão convertidos na representação solicitada pelo aplicativo de recebimento, antes que os dados sejam colocados no parâmetro **Buffer**:

- O campo `Format` nas informações de controle na mensagem identifica a estrutura de dados do aplicativo e os campos `CodedCharSetId` e `Encoding` nas informações de controle na mensagem especificam seu identificador e codificação do conjunto de caracteres.
- O aplicativo que emite a chamada `MQGET` especifica nos campos `CodedCharSetId` e `Encoding` no parâmetro **MsgDesc** o identificador do conjunto de caracteres e a codificação para o qual converter os dados da mensagem do aplicativo.

Quando a conversão dos dados da mensagem for necessária, a conversão será executada pelo próprio gerenciador de fila ou por uma saída gravada pelo usuário, dependendo do valor do campo `Format` nas informações de controle na mensagem:

- Os seguintes nomes de formato são formatos que são convertidos pelo gerenciador de filas; esses formatos são chamados de formatos “integrados”:
  - `MQFMT_ADMIN`
  - `MQFMT_CICS` ( z/OS apenas)
  - `MQFMT_COMMAND_1`
  - `MQFMT_COMMAND_2`
  - `MQFMT_DEAD_LETTER_HEADER`
  - `MQFMT_DIST_HEADER`
  - `MQFMT_EVENT` versão 1
  - `MQFMT_EVENT` versão 2 (somente z/OS).
  - `MQFMT_IMS`
  - `MQFMT_IMS_VAR_STRING`
  - `MQFMT_MD_EXTENSION`
  - `MQFMT_PCF`
  - `MQFMT_REF_MSG_HEADER`
  - `MQFMT_RF_HEADER`
  - `MQFMT_RF_HEADER_2`
  - `MQFMT_STRING`
  - `MQFMT_TRIGGER`
  - `MQFMT_WORK_INFO_HEADER` (apenas z/OS )
  - `MQFMT_XMIT_Q_HEADER`
- O nome do formato `MQFMT_NONE` é um valor especial que indica que a natureza dos dados na mensagem é indefinida. Como consequência, o gerenciador de filas não tenta a conversão quando a mensagem é recuperada da fila.

**Nota:** Se `MQGMO_CONVERT` for especificado na chamada `MQGET` para uma mensagem que possui um nome de formato `MQFMT_NONE` e o conjunto de caractere ou a codificação da mensagem for diferente daquele especificado no parâmetro **MsgDesc** , a mensagem será retornada no parâmetro **Buffer** (assumindo que não haja outros erros), mas a chamada será concluída com o código de conclusão `MQCC_WARNING` e o código de razão `MQRC_FORMAT_ERROR`.

Você pode usar `MQFMT_NONE` quando a natureza dos dados da mensagem significa que ele não requerem conversão ou quando os aplicativos de envio e recebimento acordaram entre eles o formato no qual enviar os dados da mensagem.

- Todos os outros nomes de formato transmitem a mensagem para uma saída gravada pelo usuário para conversão. A saída possui o mesmo nome que o formato, à parte das adições específicas do ambiente. Os nomes de formato especificados pelo usuário não devem começar com as letras `IBM MQ`

Consulte [“saída de conversão de dados”](#) na página 939 para obter detalhes da saída de conversão de dados.

Os dados do usuário na mensagem podem ser convertidos entre quaisquer conjuntos de caracteres e codificações suportados. No entanto, lembre-se de que, se a mensagem contiver uma ou mais estruturas de cabeçalho IBM MQ, a mensagem não poderá ser convertida de ou para um conjunto de caracteres que tenha caracteres de byte duplo ou multibyte para qualquer um dos caracteres válidos em nomes de filas. O código de razão MQR\_C\_SOURCE\_CCSD\_ERROR ou MQR\_C\_TARGET\_CCSD\_ERROR resultará se isto for tentado e a mensagem será retornada não convertida. O conjunto de caracteres Unicode UTF-16 é um exemplo desse conjunto de caracteres.

No retorno de MQGET, o código de razão a seguir indica que a mensagem foi convertida com êxito:

- MQR\_NONE

O código de razão a seguir indica que a mensagem pode ter sido convertida com êxito; o aplicativo deve verificar os campos CodedCharSetId e Encoding no parâmetro **MsgDesc** para descobrir:

- MQR\_TRUNCATED\_MSG\_ACCEPTED

Todos os outros códigos de razão indicam que a mensagem não foi convertida.

**Nota:** A interpretação desse código de razão é verdadeira para as conversões executadas por uma saída gravada pelo usuário somente se a saída estiver em conformidade com as diretrizes de processamento descritas em [“saída de conversão de dados”](#) na página 939

9. Ao usar a interface orientada a objetos para receber mensagens, é possível escolher não especificar um buffer para conter os dados da mensagem para uma chamada MQGET. Quando você obtém uma mensagem usando um aplicativo orientado a objetos sem restringir o tamanho do buffer de mensagem de recebimento, o aplicativo não falha com MQR\_CONVERTED\_MSG\_TOO\_BIG e recebe a mensagem convertida. Isto é verdadeiro para os seguintes ambientes:

- .NET, incluindo aplicativos totalmente gerenciados
- C++
- Java ( IBM MQ classes for Java )

**Nota:** Para todos os clientes, se o valor de `sharingConversations` for zero e o buffer for muito pequeno para receber a mensagem convertida, a mensagem não convertida será retornada, com o código de razão MQR\_CONVERTED\_MSG\_TOO\_BIG. Para obter mais informações sobre o `sharingConversations`, consulte [Usando conversas de compartilhamento em um aplicativo cliente](#)

10. Para os formatos integrados, o gerenciador de filas pode executar a *conversão padrão* de cadeias de caracteres na mensagem quando a opção MQGMO\_CONVERT for especificada. A conversão padrão permite que o gerenciador de filas use um conjunto de caracteres padrão especificado pela instalação que aproxima o conjunto de caracteres real, ao converter dados de cadeia. Como resultado, a chamada MQGET pode ser bem-sucedida com o código de conclusão MQCC\_OK, em vez de concluir com MQCC\_WARNING e o código de razão MQR\_C\_SOURCE\_CCSD\_ERROR ou MQR\_C\_TARGET\_CCSD\_ERROR.

**Nota:** O resultado de usar um conjunto de caracteres aproximado para converter dados de cadeia é que alguns caracteres podem ser convertidos incorretamente. Para evitar isto, use caracteres na cadeia que sejam comuns ao conjunto de caracteres real e ao conjunto de caracteres padrão.

A conversão padrão se aplica aos dados da mensagem do aplicativo e aos campos de caractere nas estruturas MQMD e MQMDE:

- A conversão padrão dos dados da mensagem do aplicativo ocorre apenas quando todas as seguintes instruções são verdadeiras:
  - O aplicativo especifica MQGMO\_CONVERT.
  - A mensagem contém dados que devem ser convertidos a partir ou para um conjunto de caracteres que não é suportado.
  - A conversão padrão foi ativada quando o gerenciador de filas foi instalado ou reiniciado.
- A conversão padrão dos campos de caractere nas estruturas MQMD e MQMDE ocorre conforme necessário, desde que a conversão padrão esteja ativada para o gerenciador de filas. A conversão é

executada mesmo se a opção MQGMO\_CONVERT não for especificada pelo aplicativo na chamada MQGET.

11. Para a linguagem de programação Visual Basic, os seguintes pontos se aplicam:

- Se o tamanho do parâmetro **Buffer** for menor que o comprimento especificado pelo parâmetro **BufferLength**, a chamada falhará com código de razão MQRC\_STORAGE\_NOT\_AVAILABLE.
- O parâmetro **Buffer** é declarado como sendo do tipo `String`. Se os dados a serem recuperados da fila não forem do tipo `String`, use a chamada MQGETAny no lugar de MQGET.

A chamada MQGETAny possui os mesmos parâmetros que a chamada MQGET, exceto que o parâmetro **Buffer** é declarado como sendo do tipo `Any`, permitindo que qualquer tipo de dado seja recuperado. No entanto, isso significa que **Buffer** não pode ser verificado para assegurar que tenha pelo menos **BufferLength** bytes de tamanho.

12. Nem todas as opções MQGET são suportadas quando a leitura antecipada é ativada. A tabela a seguir indica quais opções são permitidas e se elas podem ser alteradas entre chamadas MQGET.

	Permitido quando a leitura antecipada está ativada e pode ser alterada entre chamadas MQGET	Permitido quando a leitura antecipada está ativada, mas não pode ser alterada entre chamadas MQGET <sup>a</sup>	Opções MQGET que não são permitidas quando a leitura antecipada está ativada <sup>b</sup> .
Valores de MQGET MD	MsgId <sup>c</sup> CorrelId <sup>c</sup>	Codificação CodedCharSetId	
Opções MQGET MQGMO	MQGMO_WAIT MQGMO_NO_WAIT MQGMO_FAIL_IF QUIESCING MQGMO_BROWSE_FIRST <sup>d</sup> MQGMO_BROWSE_NEXT <sup>d</sup> MQGMO_BROWSE_MESSAGE_UNDER_CURSOR <sup>d</sup>	MQGMO_SYNCPOINT_IF_PERSISTENT MQGMO_NO_SYNCPOINT MQGMO_ACCEPT_TRUNCATED_MSG MQGMO_CONVERT MQGMO_LOGICAL_ORDER MQGMO_COMPLETE_MSG MQGMO_ALL_MSGS_AVAILABLE MQGMO_ALL_SEGMENTS_AVAILABLE MQGMO_MARK_BROWSE_HANDLE MQGMO_MARK_BROWSE_CO_OP MQGMO_UNMARK_BROWSE_CO_OP MQGMO_UNMARK_BROWSE_HANDLE MQGMO_UNMARKED_BROWSE_MSG MQGMO_PROPERTIES_FORCE_MQRFH2 MQGMO_NO_PROPERTIES MQGMO_PROPERTIES_IN_HANDLE MQGMO_PROPERTIES_COMPATIBILITY	MQGMO_SET_SIGNAL MQGMO_SYNCPOINT MQGMO_MARK_SKIP_BACKOUT MQGMO_MSG_UNDER_CURSOR <sup>d</sup> MQGMO_LOCK MQGMO_UNLOCK
Valores de MQGMO		MsgHandle	

- Se estas opções forem alteradas entre chamadas MQGET, um código de razão MQRC\_OPTIONS\_CHANGED será retornado.
  - Se estas opções forem especificadas na primeira chamada MQGET, a leitura antecipada é desativada. Se essas opções forem especificadas em uma chamada MQGET subsequente, um código de razão MQRC\_OPTIONS\_ERROR será retornado.
  - Os aplicativos clientes precisam estar cientes que, se os valores de `MsgId` e `CorrelId` forem alterados entre chamadas MQGET, mensagens com os valores anteriores poderão já ter sido enviadas ao cliente e permanecerão no buffer de leitura antecipada do cliente até serem consumidos (ou limpos automaticamente).
  - A primeira chamada MQGET determina se mensagens devem ser navegadas ou obtidas a partir de uma fila quando a leitura antecipada está ativada. Se o aplicativo tentar usar uma combinação de navegação e obtenção, um código de razão MQRC\_OPTIONS\_CHANGED será retornado.
  - MQGMO\_MSG\_UNDER\_CURSOR não é possível com a leitura antecipada. As mensagens podem ser navegadas ou obtidas quando a leitura antecipada é ativada, mas não uma combinação de ambos.
13. Os aplicativos podem receber destrutivamente mensagens não confirmadas somente se essas mensagens foram colocadas na mesma unidade de trabalho local que o recebimento. Os aplicativos não podem receber mensagens não confirmadas não destrutivamente.
14. As mensagens sob um cursor de navegação podem ser recuperadas em uma unidade de trabalho. Não é possível recuperar uma mensagem não confirmada desta maneira.

## Chamada C

```
MQGET (Hconn, Hobj, &MsgDesc, &GetMsgOpts, BufferLength, Buffer,  
&DataLength, &CompCode, &Reason);
```

Declare os parâmetros conforme a seguir:

```
MQHCONN  Hconn;           /* Connection handle */  
MQHOBJ   Hobj;           /* Object handle */  
MQMD     MsgDesc;       /* Message descriptor */  
MQGMO    GetMsgOpts;    /* Options that control the action of MQGET */  
MQLONG   BufferLength;   /* Length in bytes of the Buffer area */  
MQBYTE   Buffer[n];     /* Area to contain the message data */  
MQLONG   DataLength;    /* Length of the message */  
MQLONG   CompCode;     /* Completion code */  
MQLONG   Reason;       /* Reason code qualifying CompCode */
```

## Chamada COBOL

```
CALL 'MQGET' USING HCONN, HOBJ, MSGDESC, GETMSGOPTS, BUFFERLENGTH,  
BUFFER, DATALENGTH, COMPCODE, REASON.
```

Declare os parâmetros conforme a seguir:

```
** Connection handle  
01 HCONN          PIC S9(9) BINARY.  
** Object handle  
01 HOBJ          PIC S9(9) BINARY.  
** Message descriptor  
01 MSGDESC.  
   COPY CMQMDV.  
** Options that control the action of MQGET  
01 GETMSGOPTS.  
   COPY CMQGMV.  
** Length in bytes of the BUFFER area  
01 BUFFERLENGTH PIC S9(9) BINARY.  
** Area to contain the message data  
01 BUFFER       PIC X(n).  
** Length of the message  
01 DATALENGTH  PIC S9(9) BINARY.  
** Completion code  
01 COMPCODE     PIC S9(9) BINARY.  
** Reason code qualifying COMPCODE  
01 REASON       PIC S9(9) BINARY.
```

## Chamada PL/I

```
call MQGET (Hconn, Hobj, MsgDesc, GetMsgOpts, BufferLength, Buffer,  
DataLength, CompCode, Reason);
```

Declare os parâmetros conforme a seguir:

```
dc1 Hconn          fixed bin(31); /* Connection handle */  
dc1 Hobj          fixed bin(31); /* Object handle */  
dc1 MsgDesc       like MQMD;     /* Message descriptor */  
dc1 GetMsgOpts    like MQGMO;    /* Options that control the action of  
MQGET */  
dc1 BufferLength   fixed bin(31); /* Length in bytes of the Buffer  
area */  
dc1 Buffer         char(n);       /* Area to contain the message data */  
dc1 DataLength    fixed bin(31); /* Length of the message */  
dc1 CompCode      fixed bin(31); /* Completion code */  
dc1 Reason        fixed bin(31); /* Reason code qualifying CompCode */
```

## Chamada do Assembler de Alto Nível

```
CALL MQGET, (HCONN, HOBJ, MSGDESC, GETMSGOPTS, BUFFERLENGTH,
            BUFFER, DATALENGTH, COMPCODE, REASON)
```

Declare os parâmetros conforme a seguir:

HCONN	DS	F	Connection handle
HOBJ	DS	F	Object handle
MSGDESC	CMQMDA	,	Message descriptor
GETMSGOPTS	CMQGMOA	,	Options that control the action of MQGET
BUFFERLENGTH	DS	F	Length in bytes of the BUFFER area
BUFFER	DS	CL(n)	Area to contain the message data
DATALENGTH	DS	F	Length of the message
COMPCODE	DS	F	Completion code
REASON	DS	F	Reason code qualifying COMPCODE

## Chamada do Visual Basic

```
MQGET Hconn, Hobj, MsgDesc, GetMsgOpts, BufferLength, Buffer,
DataLength, CompCode, Reason
```

Declare os parâmetros conforme a seguir:

Dim Hconn	As Long	'Connection handle'
Dim Hobj	As Long	'Object handle'
Dim MsgDesc	As MQMD	'Message descriptor'
Dim GetMsgOpts	As MQGMO	'Options that control the action of MQGET'
Dim BufferLength	As Long	'Length in bytes of the Buffer area'
Dim Buffer	As String	'Area to contain the message data'
Dim DataLength	As Long	'Length of the message'
Dim CompCode	As Long	'Completion code'
Dim Reason	As Long	'Reason code qualifying CompCode'

## MQINQ-Consultar atributos do objeto.

A chamada MQINQ retorna uma matriz de números inteiros e um conjunto de cadeias de caracteres que contêm os atributos de um objeto

Os seguintes tipos de objeto são válidos:

- Gerenciador de filas
- Fila
- Lista de Nomes
- Definição de processo

## Sintaxe

```
MQINQ (Hconn, Hobj, SelectorCount, Seletores, IntAttrCount, IntAttrs, CharAttrLength, CharAttrs,
CompCode, Reason)
```

## Parâmetros

### Hconn

Tipo: MQHCONN -entrada

Esta manipulação representa a conexão ao gerenciador de filas. O valor de *Hconn* foi retornado por uma chamada anterior de MQCONN ou MQCONNX

Em z/OS para aplicativos CICS, a chamada MQCONN pode ser omitida e o seguinte valor especificado para *Hconn*:

## **MQHC\_DEF\_HCONN**

Manipulação de conexões padrão.

### **Hobj**

Tipo: MQHOBJ -entrada

Essa manipulação representa o objeto (de qualquer tipo) com atributos necessários. O identificador deve ser retornado por uma chamada MQOPEN anterior que especificou a opção MQ00\_INQUIRE .

### **SelectorCount**

Tipo: MQLONG -entrada

Essa é a contagem de seletores fornecidos na matriz *Selectors* . É o número de atributo que deve ser retornado. Zero é um valor válido. O número máximo permitido é 256..

### **Seletores**

Tipo: MQLONG x *SelectorCount* -entrada

Esta é uma matriz de seletores de atributo **SelectorCount** ; cada seletor identifica um atributo (número inteiro ou caractere) com um valor que é necessário.

Cada seletor deve ser válido para o tipo de objeto que *Hobj* representa, caso contrário, a chamada falhará com código de conclusão MQCC\_FAILED e código de razão MQRC\_SELECTOR\_ERROR.

No caso especial de filas:

- Se o seletor não for válido para filas de qualquer tipo, a chamada falhará com código de conclusão MQCC\_FAILED e código de razão MQRC\_SELECTOR\_ERROR.
- Se o seletor se aplicar apenas a filas de tipos diferentes do tipo do objeto, a chamada será bem-sucedida com código de conclusão MQCC\_WARNING e código de razão MQRC\_SELECTOR\_NOT\_FOR\_TYPE.
- Se a fila sendo consultada for uma fila de clusters, os seletores que são válidos dependem de como a fila foi resolvida; consulte “Observações de Uso” na página 746 para obter detalhes adicionais.

É possível especificar os seletores em qualquer ordem. Os valores de atributo que correspondem a seletores de atributo de número inteiro (seletores MQIA\_\*) são retornados em *IntAttrs* na mesma ordem em que esses seletores ocorrem em *Selectors*... Os valores de atributo que correspondem aos seletores de atributo de caractere (seletores MQCA\_\*) são retornados em *CharAttrs* na mesma ordem em que esses seletores ocorrem.. Os seletores MQIA\_\* podem ser intercalados com os seletores MQCA\_\* ; somente a ordem relativa dentro de cada tipo é importante

### **Nota:**

1. Os seletores de atributo de número inteiro e de caractere são alocados em dois intervalos diferentes; os seletores MQIA\_\* residem no intervalo de MQIA\_FIRST a MQIA\_LAST e os seletores MQCA\_\* no intervalo de MQCA\_FIRST a MQCA\_LAST.

Para cada intervalo, as constantes MQIA\_LAST\_USED e MQCA\_LAST\_USED definem o valor mais alto aceito pelo gerenciador de filas.

2. Se todos os seletores MQIA\_\* ocorrerem primeiro, os mesmos números de elementos poderão ser usados para tratar elementos correspondentes nas matrizes *Selectors* e *IntAttrs* .
3. Se o parâmetro **SelectorCount** for zero, *Selectors* não será referido. Nesse caso, o endereço do parâmetro transmitido por programas gravados em C ou assembler S/390 pode ser nulo.

Os atributos que podem ser consultados são listados nas tabelas a seguir: Para os seletores MQCA\_\* , a constante que define o comprimento em bytes da cadeia resultante em *CharAttrs* é fornecida entre parênteses.

As tabelas a seguir listam os seletores, por objeto, em ordem alfabética, conforme a seguir:

- Seletores de atributo do [Tabela 549 na página 732](#) MQINQ para filas
- Seletores de atributo do [Tabela 550 na página 734](#) MQINQ para listas de nomes
- seletores de atributo do [Tabela 551 na página 735](#) MQINQ para definições de processo

- seletores de atributo do [Tabela 552 na página 735 MQINQ](#) para o gerenciador de filas

Todos os seletores são suportados em todas as plataformas IBM MQ , exceto quando indicado na coluna **Nota** , conforme a seguir:

**Não z/OS**

Suportado em todas as plataformas **exceto** z/OS

**z/OS**

Suportado **apenas** em z/OS

<i>Tabela 549. Seletores de atributo MQINQ para filas</i>			
<b>Seletor</b>	<b>Comprimento do campo</b>	<b>Descrição</b>	<b>Nota</b>
MQCA_ALTERATION_DATE	MQ_DATE_LENGTH	Data da alteração mais recente	
MQCA_ALTERATION_TIME	MQ_TIME_LENGTH	Hora da alteração mais recente	
MQCA_BACKOUT_REQ_Q_NAME	MQ_Q_NAME_LENGTH	Nome de enfileiramento de restauração excessivo	
MQCA_BASE_Q_NAME	MQ_Q_NAME_LENGTH	Nome da fila para a qual o alias é resolvido..	
MQCA_CF_STRUC_NAME	MQ_CF_STRUC_NAME_LENGTH	Nome da estrutura de recursos de acoplamento	z/OS
MQCA_CLUS_CHL_NAME	MQ_CHANNEL_NAME_LENGTH	Nome do canal do emissor de clusters que usa essa fila como uma fila de transmissão	
MQCA_CLUSTER_NAME	MQ_CLUSTER_NAME_LENGTH	Nome do cluster	
MQCA_CLUSTER_NAMELIST	MQ_NAMELIST_NAME_LENGTH	Lista de nomes do cluster	
MQCA_CREATION_DATE	MQ_CREATION_DATE_LENGTH	Data de criação da fila.	
MQCA_CREATION_TIME	MQ_CREATION_TIME_LENGTH	Horário de criação da fila..	
MQCA_CUSTOM	MQ_CUSTOM_LENGTH	O atributo customizado para novos recursos	
MQCA_INITIATION_Q_NAME	MQ_Q_NAME_LENGTH	Nome da Fila de Inicialização	
MQCA_PROCESS_NAME	MQ_PROCESS_NAME_LENGTH	Nome da definição de processo:	
MQCA_Q_DESC	MQ_Q_DESC_LENGTH	Descrição da fila	
MQCA_Q_NAME	MQ_Q_NAME_LENGTH	Nome da fila	
MQCA_REMOTE_Q_MGR_NAME	MQ_Q_MGR_NAME_LENGTH	Nome do gerenciador de fila remoto	
MQCA_REMOTE_Q_NAME	MQ_Q_NAME_LENGTH	Nome da fila remota conforme conhecido no gerenciador de filas remotas	
MQCA_STORAGE_CLASS	MQ_STORAGE_CLASS_LENGTH	Nome da classe de armazenamento	z/OS
MQCA_TRIGGER_DATA	MQ_TRIGGER_DATA_LENGTH	Dados de Ativação	



Tabela 549. Seletores de atributo MQINQ para filas (continuação)

<b>Seletor</b>	<b>Comprimento do campo</b>	<b>Descrição</b>	<b>Nota</b>
MQCA_XMIT_Q_NAME	MQ_Q_NAME_LENGTH	Nome da fila de transmissão	
MQIA_ACCOUNTING_Q	MQLONG	Controla a coleta de dados contábeis para fila	Não z/OS
MQIA_BACKOUT_THRESHOLD	MQLONG	Limite de Recuperação	
MQIA_CLWL_Q_PRIORITY	MQLONG	Prioridade da fila	
MQIA_CLWL_Q_RANK	MQLONG	Classificação da fila.	
MQIA_CLWL_USEQ	MQLONG	Utilizar filas remotas	
MQIA_CURRENT_Q_DEPTH	MQLONG	Número de mensagens na fila	
MQIA_DEF_BIND	MQLONG	Ligação padrão	
MQIA_DEF_INPUT_OPEN_OPTION	MQLONG	Opção de abertura para entrada padrão	
MQIA_DEF_PERSISTENCE	MQLONG	Persistência padrão de mensagem	
MQIA_DEF_PRIORITY	MQLONG	Prioridade padrão da mensagem	
MQIA_DEFINITION_TYPE	MQLONG	Tipo de definição de fila	
MQIA_DIST_LISTS	MQLONG	Suporte à lista de distribuição	Não z/OS
MQIA_HARDEN_GET_BACKOUT	MQLONG	Se a contagem de restaurações deve ser prejudicada	
MQIA_INDEX_TYPE	MQLONG	Tipo de índice mantido para a fila	z/OS
MQIA_INHIBIT_GET	MQLONG	Se as operações get são permitidas	
MQIA_INHIBIT_PUT	MQLONG	Se as operações put são permitidas	
MQIA_MAX_MSG_LENGTH	MQLONG	Comprimento máximo de mensagem	
MQIA_MAX_Q_DEPTH	MQLONG	Número máximo de mensagens permitidas na fila	
MQIA_MSG_DELIVERY_SEQUENCE	MQLONG	Se a prioridade da mensagem é relevante	
MQIA_NPM_CLASS	MQLONG	Nível de confiabilidade para mensagens não persistentes	
MQIA_OPEN_INPUT_COUNT	MQLONG	Número de chamadas MQOPEN que possuem a fila aberta para entrada	
MQIA_OPEN_OUTPUT_COUNT	MQLONG	Número de chamadas MQOPEN que possuem a fila aberta para saída	
MQIA_PROPERTY_CONTROL	MQLONG	atributo de controle de propriedade	
MQIA_Q_DEPTH_HIGH_EVENT	MQLONG	Atributo de controle para eventos altos de profundidade da fila	Não z/OS
MQIA_Q_DEPTH_HIGH_LIMIT	MQLONG	Limite alto para profundidade da fila	Não z/OS
MQIA_Q_DEPTH_LOW_EVENT	MQLONG	Atributo de controle para eventos de baixa profundidade da fila	Não z/OS

<i>Tabela 549. Seletores de atributo MQINQ para filas (continuação)</i>			
<b>Seletor</b>	<b>Comprimento do campo</b>	<b>Descrição</b>	<b>Nota</b>
MQIA_Q_DEPTH_LOW_LIMIT	MQLONG	Limite baixo para profundidade da fila	Não z/OS
MQIA_Q_DEPTH_MAX_EVENT	MQLONG	Atributo de controle para eventos máximos de profundidade da fila	Não z/OS
MQIA_Q_SERVICE_INTERVAL	MQLONG	Limite para intervalo de serviço de fila	Não z/OS
MQIA_Q_SERVICE_INTERVAL_EVENT	MQLONG	Atributo de controle para eventos de intervalo de serviço de fila	Não z/OS
MQIA_Q_TYPE	MQLONG	Tipo de fila	
MQIA_QSG_DISP	MQLONG	Disposição do Grupo de Compartilhamento de Fila	z/OS
MQIA_RETENTION_INTERVAL	MQLONG	Intervalo de retenção da fila	
MQIA_SCOPE	MQLONG	Escopo de definição de fila..	Não z/OS
MQIA_SHAREABILITY	MQLONG	Se a fila pode ser compartilhada para entrada	
MQIA_STATISTICS_Q	MQLONG	Controla a coleta de dados estatísticos para a fila	Não z/OS
MQIA_TRIGGER_CONTROL	MQLONG	Controle de ativação	
MQIA_TRIGGER_DEPTH	MQLONG	Profundidade de ativação	
MQIA_TRIGGER_MSG_PRIORITY	MQLONG	Prioridade da mensagem limite para acionadores	
MQIA_TRIGGER_TYPE	MQLONG	Tipo de ativação	
MQIA_USAGE	MQLONG	Uso	

<i>Tabela 550. Seletores de atributo MQINQ para listas de nomes</i>			
<b>Seletor</b>	<b>Comprimento do campo</b>	<b>Descrição</b>	<b>Nota</b>
MQCA_ALTERATION_DATE	MQ_DATE_LENGTH	Data da alteração mais recente	
MQCA_ALTERATION_TIME	MQ_TIME_LENGTH	Tempo de alteração mais recente	
MQCA_NAMELIST_DESC	MQ_NAMELIST_DESC_LENGTH	Descrição da lista de nomes:.	
MQCA_NAMELIST_NAME	MQ_NAMELIST_NAME_LENGTH	Nome do objeto da lista de nomes	
MQIA_NAMELIST_TYPE	MQLONG	Tipo de Lista de Nomes	z/OS
MQCA_NAMES	MQ_Q_NAME_LENGTH <i>x Number of names in the list</i>	Nomes na lista de nomes	
MQIA_NAME_COUNT	MQLONG	Número de nomes na lista de nomes	
MQIA_QSG_DISP	MQLONG	Disposição do Grupo de Compartilhamento de Fila	z/OS

<i>Tabela 551. seletores de atributos MQINQ para definições de processo</i>			
<b>Seletor</b>	<b>Comprimento do campo</b>	<b>Descrição</b>	<b>Nota</b>
MQCA_ALTERATION_DATE	MQ_DATE_LENGTH	Data da alteração mais recente	
MQCA_ALTERATION_TIME	MQ_TIME_LENGTH	Tempo de alteração mais recente	
MQCA_APPL_ID	MQ_PROCESS_APPL_ID_LENGTH	Identificador do aplicativo	
MQCA_ENV_DATA	MQ_PROCESS_ENV_DATA_LENGTH	Dados do ambiente	
MQCA_PROCESS_DESC	MQ_PROCESS_DESC_LENGTH	Descrição de definição de processo	
MQCA_PROCESS_NAME	MQ_PROCESS_NAME_LENGTH	Nome da definição de processo:	
MQCA_USER_DATA	MQ_PROCESS_USER_DATA_LENGTH	Dados do usuário	
MQIA_APPL_TYPE	MQLONG	Tipo de aplicativo	
MQIA_QSG_DISP	MQLONG	Disposição do Grupo de Compartilhamento de Fila	z/OS

<i>Tabela 552. seletores de atributo MQINQ para o gerenciador de filas</i>			
<b>Seletor</b>	<b>Comprimento do campo</b>	<b>Descrição</b>	<b>Nota</b>
MQCA_ALTERATION_DATE	MQ_DATE_LENGTH	Data da alteração mais recente	
MQCA_ALTERATION_TIME	MQ_TIME_LENGTH	Tempo de alteração mais recente	
MQCA_CHANNEL_AUTO_DEF_EXIT	MQ_EXIT_NAME_LENGTH	Nome da saída da definição de canal automática	
MQCA_CHINIT_SERVICE_PARM		Reservado para uso pelo IBM	
MQCA_CLUSTER_WORKLOAD_DATA	MQ_EXIT_DATA_LENGTH	Dados passados para a saída de carga de trabalho do cluster..	
MQCA_CLUSTER_WORKLOAD_EXIT	MQ_EXIT_NAME_LENGTH	Nome da saída de carga de trabalho do cluster..	
MQCA_COMMAND_INPUT_Q_NAME	MQ_Q_NAME_LENGTH	Nome da fila de entrada do comando do sistema.	
MQCA_CUSTOM	MQ_CUSTOM_LENGTH	O atributo customizado para novos recursos	
MQCA_DEAD_LETTER_Q_NAME	MQ_Q_NAME_LENGTH	Nome da fila de mensagens não entregues	
MQCA_DEF_XMIT_Q_NAME	MQ_Q_NAME_LENGTH	Nome da Fila de Transmissão Padrão	
MQCA_DNS_GROUP	MQ_DNS_GROUP_NAME_LENGTH	Nome do grupo para o listener TCP que manipula transmissões de entrada para o grupo de filas compartilhadas para junção. O nome se aplica ao usar o Workload Manager Dynamic Domain Name Services.	z/OS

Tabela 552. seletores de atributo MQINQ para o gerenciador de filas (continuação)


Seletor	Comprimento do campo	Descrição	Nota
MQCA_IGQ_USER_ID	MQ_USER_ID_LENGTH	Identificador de usuário de enfileiramento intragrupo	z/OS
 MQCA_INITIAL_KEY	MQ_INITIAL_KEY_LENGTH	Chave inicial para o sistema de proteção de senha	Retorna *** ** se não estiver em branco ou em branco se a chave inicial padrão estiver em uso
MQCA_INSTALLATION_DESC	MQ_INSTALLATION_DESC_LENGTH	Descrição da instalação associada	Não z/OS · NÃO IBM i
MQCA_INSTALLATION_NAME	MQ_INSTALLATION_NAME_LENGTH	Nome da instalação associada ao gerenciador de filas	Não z/OS · NÃO IBM i
MQCA_INSTALLATION_PATH	MQ_INSTALLATION_PATH_LENGTH	Caminho no qual o IBM MQ associado está instalado	Não z/OS · NÃO IBM i
MQCA_LU_GROUP_NAME	MQ_LU_NAME_LENGTH	Nome da LU genérica para o listener da LU 6.2 que manipula transmissões de entrada para o grupo de filas compartilhadas a ser usado	z/OS

Tabela 552. seletores de atributo MQINQ para o gerenciador de filas (continuação)

<b>Seletor</b>	<b>Comprimento do campo</b>	<b>Descrição</b>	<b>Nota</b>
MQCA_LU_NAME	MQ_LU_NAME_LENGTH	Nome da LU a ser utilizado para transmissões de LU de saída 6.2 Configure esse nome para a mesma LU que o listener usa para transmissões de entrada..	z/OS
MQCA_LU62_ARM_SUFFIX	MQ_ARM_SUFFIX_LENGTH	Sufixo do SYS1 . PARMLIB membro APPCPM <i>xx</i> , que nomeia o LUADD para este inicializador de canais	z/OS
MQCA_PARENT	MQ_Q_MGR_NAME_LENGTH	Nome de um gerenciador de filas conectado hierarquicamente que é nomeado como o pai deste gerenciador de fila	
MQCA_Q_MGR_DESC	MQ_Q_MGR_DESC_LENGTH	Descrição do gerenciador de filas	
MQCA_Q_MGR_IDENTIFIER	MQ_Q_MGR_IDENTIFIER_LENGTH	Identificador do gerenciador de filas (H).	
MQCA_Q_MGR_NAME	MQ_Q_MGR_NAME_LENGTH	Nome do gerenciador de filas locais	
MQCA_QSG_NAME	MQ_QSG_NAME_LENGTH	Nome do grupo de filas compartilhadas	z/OS
MQCA_REPOSITORY_NAME	MQ_CLUSTER_NAME_LENGTH	Nome do cluster para o qual o gerenciador de filas fornece serviços de repositório	
MQCA_REPOSITORY_NAMELIST	MQ_NAMELIST_NAME_LENGTH	Nome do objeto da lista de nomes contendo nomes de clusters para os quais o gerenciador de filas fornece serviços de repositório	

Tabela 552. seletores de atributo MQINQ para o gerenciador de filas (continuação)


Seletor	Comprimento do campo	Descrição	Nota
 MQCA_SSL_KEY_REPO_PASSWORD	MQ_SSL_ENCRYPT_KEY_REPO_PWD_LEN	Senha do repositório de chaves	Retorna *** se não estiver em branco ou em branco se não estiver configurado. Criptografado quando configurado antes do armazenamento.
MQCA_TCP_NAME	MQ_TCP_NAME_LENGTH	Nome do sistema TCP/IP que você está usando	z/OS
MQIA_ACCOUNTING_CONN_OVERRIDE	MQLONG	Substituir configurações de contabilidade	Não z/OS
MQIA_ACCOUNTING_INTERVAL	MQLONG	Com que frequência gravar registros de contabilidade intermediários	Não z/OS
MQIA_ACCOUNTING_MQI	MQLONG	Controla a coleta de informações de contabilidade para dados de MQI	Não z/OS
MQIA_ACCOUNTING_Q	MQLONG	Controla a coleta de informações de contabilidade para filas	Não z/OS
MQIA_ACTIVE_CHANNELS	MQLONG	Número máximo de canais que podem estar ativos a qualquer momento	z/OS

Tabela 552. seletores de atributo MQINQ para o gerenciador de filas (continuação)

<b>Seletor</b>	<b>Comprimento do campo</b>	<b>Descrição</b>	<b>Nota</b>
MQIA_ADOPTNEWMCA_CHECK	MQLONG	Elementos que são verificados para determinar se devem adotar um MCA. A verificação é executada quando um novo canal de entrada é detectado com o mesmo nome que um MCA que já está ativo.	z/OS
MQIA_ADOPTNEWMCA_INTERVAL	MQLONG	Quantidade de tempo, em segundos, que o novo canal aguarda o canal órfão terminar	Não z/OS
MQIA_ADOPTNEWMCA_TYPE	MQLONG	Se deve-se reiniciar uma instância órfã de um MCA de um tipo de canal específico automaticamente quando uma nova solicitação do canal de entrada correspondente aos parâmetros AdoptNewMCA Check for detectada	z/OS
MQIA_AUTHORITY_EVENT	MQLONG	Atributo de controle para eventos de autoridade..	Não z/OS
MQIA_BRIDGE_EVENT	MQLONG	Atributo de controle para eventos de ponte IMS	z/OS
MQIA_CHANNEL_AUTO_DEF	MQLONG	Atributo de controle para definição de canal automático	Não z/OS
MQIA_CHANNEL_AUTO_DEF_EVENT	MQLONG	Atributo de controle para eventos de definição de canal automática	Não z/OS
MQIA_CHANNEL_EVENT	MQLONG	Atributo de controle para eventos do canal.	
MQIA_CHINIT_ADAPTERS	MQLONG	Número de subtarefas do adaptador a serem usadas para processar chamadas IBM MQ	z/OS
MQIA_CHINIT_DISPATCHERS	MQLONG	Número de dispatchers a serem usados para o inicializador de canais	z/OS
MQIA_CHINIT_TRACE_AUTO_START	MQLONG	Se o rastreamento do inicializador de canais deve ser iniciado automaticamente	z/OS
MQIA_CHINIT_TRACE_TABLE_SIZE	MQLONG	Tamanho do espaço para dados de rastreamento (em MB) do inicializador de canais	z/OS
MQIA_CLUSTER_WORKLOAD_LENGTH	MQLONG	Comprimento da carga de trabalho do cluster	
MQIA_CLWL_MRU_CHANNELS	MQLONG	Número de canais usados mais recentemente para balanceamento de carga de trabalho do cluster	
MQIA_CLWL_USEQ	MQLONG	Utilizar filas remotas	
MQIA_CODED_CHAR_SET_ID	MQLONG	Identificador do conjunto de caracteres codificados	

Tabela 552. seletores de atributo MQINQ para o gerenciador de filas (continuação)

<b>Seletor</b>	<b>Comprimento do campo</b>	<b>Descrição</b>	<b>Nota</b>
MQIA_COMMAND_EVENT	MQLONG	Atributo de controle para eventos de comando..	
MQIA_COMMAND_LEVEL	MQLONG	Nível de comando suportado pelo gerenciador de filas	
MQIA_CONFIGURATION_EVENT	MQLONG	Atributo de controle para eventos de configuração.	Não z/OS
MQIA_DEF_CLUSTER_XMIT_Q_TYPE	MQLONG	Tipo de fila de transmissão padrão a ser utilizada para canais do emissor de clusters.	
MQIA_DIST_LISTS	MQLONG	Suporte à lista de distribuição	Não z/OS
MQIA_DNS_WLM	MQLONG	Se o listener TCP que manipula transmissões de entrada para o grupo de filas compartilhadas se registra com o Workload Manager for Dynamic Domain Name Services	z/OS
MQIA_EXPIRY_INTERVAL	MQLONG	Intervalo entre varreduras para mensagens expiradas	z/OS
MQIA_GROUP_UR	MQLONG	Atributo de controle para saber se as unidades de recuperação GROUP estão ativadas para este gerenciador de fila A disposição da unidade de recuperação GROUP está disponível apenas se o gerenciador de filas for um membro de um grupo de filas compartilhadas	z/OS
MQIA_IGQ_PUT_AUTHORITY	MQLONG	Autoridade put de enfileiramento intragrupo	z/OS
MQIA_INHIBIT_EVENT	MQLONG	Atributo de controle para eventos de inibição	Não z/OS
MQIA_INTRA_GROUP_queuing	MQLONG	Suporte ao enfileiramento intragrupo	z/OS
MQIA_LISTENER_TIMER	MQLONG	O intervalo de tempo (em segundos) entre IBM MQ tenta reiniciar o listener se APPC ou TCP/IP falhar.	z/OS
MQIA_LOCAL_EVENT	MQLONG	Atributo de controle para eventos locais	Não z/OS
MQIA_LOGGER_EVENT	MQLONG	Atributo de controle para eventos de inibição	Não z/OS
MQIA_LU62_CHANNELS	MQLONG	Número máximo de canais que podem ser atuais ou clientes que podem ser conectados, usando o protocolo de transmissão LU 6.2	z/OS



Tabela 552. seletores de atributo MQINQ para o gerenciador de filas (continuação)


Seletor	Comprimento do campo	Descrição	Nota
MQIA_MSG_MARK_BROWSE_INTERVAL	MQLONG	Intervalo de tempo (em milissegundos) após o qual o gerenciador de filas pode remover automaticamente uma marca de mensagens de navegação.   <b>Atenção:</b> Esse valor não deve ser configurado abaixo do padrão 5000.	
MQIA_MAX_CHANNELS	MQLONG	Número máximo de canais que podem ser atuais (incluindo canais de conexão do servidor com clientes conectados)	z/OS
MQIA_MAX_HANDLES	MQLONG	Número máximo de identificadores	
MQIA_MAX_MSG_LENGTH	MQLONG	Comprimento máximo de mensagem	
MQIA_MAX_PRIORITY	MQLONG	Prioridade máxima	
MQIA_MAX_UNCOMMITTED_MESSAGES	MQLONG	Número máximo de mensagens não confirmadas em uma unidade de trabalho	
MQIA_OUTBOUND_PORT_MAX	MQLONG	Com MQIA_OUTBOUND_PORT_MIN, define o intervalo de números de portas a serem usados ao ligar canais de saída	z/OS
MQIA_OUTBOUND_PORT_MIN	MQLONG	Com MQIA_OUTBOUND_PORT_MAX, define o intervalo de números de portas a serem usados ao ligar canais de saída	z/OS
MQIA_PERFORMANCE_EVENT	MQLONG	Atributo de controle para eventos de desempenho..	Não z/OS
MQIA_PLATFORM	MQLONG	Plataforma na qual o gerenciador de filas reside	
MQIA_PROT_POLICY_CAPABILITY	MQLONG	Indica se recursos de segurança do Advanced Message Security estão disponíveis para um gerenciador de fila.	
MQIA_PUBSUB_MAXMSG_RETRY_COUNT	MQLONG	O número de tentativas de reprocessar uma mensagem de comando com falha no ponto de sincronização	

Tabela 552. seletores de atributo MQINQ para o gerenciador de filas (continuação)

<b>Seletor</b>	<b>Comprimento do campo</b>	<b>Descrição</b>	<b>Nota</b>
MQIA_PUBSUB_MODE	MQLONG	Se o mecanismo de publicação / assinatura e a interface de publicação / assinatura enfileirada estão em execução Os aplicativos para publicar ou assinar usando a interface de programação de aplicativos requerem o mecanismo de publicar / assinar As filas monitoradas pela interface de publicação / assinatura enfileirada requerem que a interface de publicação / assinatura enfileirada esteja em execução.	
MQIA_PUBSUB_NP_MSG	MQLONG	Se descartar (ou manter) uma mensagem de entrada não entregue	
MQIA_PUBSUB_NP_RESP	MQLONG	Controla o comportamento de mensagens de resposta não entregues	
MQIA_PUBSUB_SYNC_PT	MQLONG	Se apenas mensagens persistentes (ou todas) são processadas no ponto de sincronização	
MQIA_QMGR_CFCONLOS	MQLONG	Especifica a ação a ser executada quando o gerenciador de filas perde a conectividade com a estrutura de administração ou quaisquer estruturas CF com CFCONLOS configurado como ASQMGR	z/OS
MQIA_RECEIVE_TIMEOUT	MQLONG	Aproximadamente quanto tempo um canal TCP/IP aguarda para receber dados, incluindo pulsações, de seu parceiro, antes de retornar ao estado inativo. O valor é numérico, qualificado por MQIA_RECEIVE_TIMEOUT_TYPE.	z/OS
MQIA_RECEIVE_TIMEOUT_MIN	MQLONG	Tempo mínimo que um canal TCP/IP aguarda para receber dados, incluindo pulsações, de seu parceiro, antes de retornar ao estado inativo	z/OS
MQIA_RECEIVE_TIMEOUT_TYPE	MQLONG	Aproximadamente quanto tempo um canal TCP/IP aguarda para receber dados, incluindo pulsações, de seu parceiro, antes de retornar ao estado inativo. MQIA_RECEIVE_TIMEOUT_TYPE é o qualificador aplicado a MQIA_RECEIVE_TIMEOUT..	z/OS
MQIA_REMOTE_EVENT	MQLONG	Atributo de controle para eventos remotos	Não z/OS
MQIA_SECURITY_CASE	MQLONG	Caso de perfis de segurança	z/OS

Tabela 552. seletores de atributo MQINQ para o gerenciador de filas (continuação)

Seletor	Comprimento do campo	Descrição	Nota
MQIA_SSL_EVENT	MQLONG	Atributo de controle para eventos do canal.	
MQIA_SSL_FIPS_REQUIRED	MQLONG	Usar apenas algoritmo certificado por FIPS para criptografia...	
MQIA_SSL_RESET_COUNT	MQLONG	Contagem de reconfiguração de chave TLS	
MQIA_START_STOP_EVENT	MQLONG	Atributo de controle para eventos de parada de início	Não z/OS
MQIA_STATISTICS_AUTO_CLUSTER	MQLONG	Controla a coleção de informações de monitoramento de estatísticas para canais do emissor de clusters	
MQIA_STATISTICS_CHANNEL	MQLONG	Controla a coleção de dados estatísticos para canais	
MQIA_STATISTICS_INTERVAL	MQLONG	Com que frequência gravar dados de monitoramento de estatísticas	Não z/OS
MQIA_STATISTICS_MQI	MQLONG	Controla a coleta de informações de monitoramento de estatísticas para o gerenciador de filas	Não z/OS
MQIA_STATISTICS_Q	MQLONG	Controla a coleta de dados estatísticos para filas	Não z/OS
MQIA_SYNCPOINT	MQLONG	disponibilidade do ponto de sincronização	
MQIA_TCP_CHANNELS	MQLONG	Número máximo de canais que podem ser atuais ou clientes que podem ser conectados usando o protocolo de transmissão TCP/IP	z/OS
MQIA_TCP_KEEP_ALIVE	MQLONG	Se o recurso TCP KEEPALIVE deve ser usado para verificar se a outra extremidade da conexão ainda está disponível	z/OS
MQIA_TCP_STACK_TYPE	MQLONG	Se o inicializador de canais pode usar apenas o espaço de endereço TCP/IP especificado em TCPNAME ou pode, opcionalmente, ligar a qualquer endereço TCP/IP selecionado	z/OS
MQIA_TRACE_ROUTE_RECORDING	MQLONG	Controla a gravação das informações de rastreamento de rotas	z/OS
MQIA_TREE_LIFE_TIME	MQLONG	Duração de tópicos não administrativos não utilizados	
MQIA_TRIGGER_INTERVAL	MQLONG	Intervalo de ativação	

**IntAttrCount**

Tipo: MQLONG -entrada

Este é o número de elementos na matriz *IntAttrs* .. Zero é um valor válido.

Se `IntAttrCount` for pelo menos o número de seletores `MQIA_*` no parâmetro **Selectors** , todos os atributos de número inteiro solicitados serão retornados.

### **IntAttrs**

Tipo: `MQLONG` x `IntAttrCount` -saída

Essa é uma matriz de valores de atributo de número inteiro `IntAttrCount` .

Os valores de atributo de número inteiro são retornados na mesma ordem dos seletores `MQIA_*` no parâmetro **Selectors** . Se a matriz contiver mais elementos do que o número de seletores `MQIA_*` , os elementos em excesso não serão alterados.

Se *Hobj* representa uma fila, mas um seletor de atributo não se aplica a esse tipo de fila, o valor específico `MQIAV_NOT_APPLICABLE` é retornado.. É retornado para o elemento correspondente na matriz `IntAttrs` .

Se o parâmetro **IntAttrCount** ou **SelectorCount** for zero, `IntAttrs` não será referido. Nesse caso, o endereço do parâmetro transmitido por programas gravados em C ou assembler S/390 pode ser nulo.

### **Comprimento de CharAttr**

Tipo: `MQLONG` -entrada

Esse é o comprimento em bytes do parâmetro **CharAttrs** .

`CharAttrLength` deve ser pelo menos a soma dos comprimentos dos atributos de caractere solicitados (consulte Seletores ).. Zero é um valor válido.

### **CharAttrs**

Tipo: `MQCHAR` x `CharAttrLength` -saída

Este é o buffer no qual os atributos de caractere são retornados, concatenados juntos O comprimento do buffer é fornecido pelo parâmetro **CharAttrLength**.

Os atributos de caractere são retornados na mesma ordem que os seletores `MQCA_*` no parâmetro **Selectors** O comprimento de cada sequência de atributos é fixo para cada atributo (consulte Seletores ) e o valor nele é preenchido à direita com espaços em branco, se necessário. É possível fornecer um buffer maior do que o necessário para conter todos os atributos de caracteres e preenchimento solicitados Os bytes além do último valor de atributo retornado permanecem inalterados.

Se *Hobj* representa uma fila, mas um seletor de atributo não se aplica a esse tipo de fila, uma sequência de caracteres que consiste inteiramente de asteriscos (\*) é retornada. O asterisco é retornado como o valor desse atributo em `CharAttrs`

Se o parâmetro `CharAttrLength` ou **SelectorCount** for zero, `CharAttrs` não será referido. Nesse caso, o endereço do parâmetro transmitido por programas gravados em C ou assembler S/390 pode ser nulo.

### **CompCode**

Tipo: `MQLONG` -saída

O código de conclusão:

#### **MQCC\_OK**

Indica conclusão bem-sucedida.

#### **MQCC\_WARNING**

Aviso (conclusão parcial).

#### **MQCC\_FAILED**

A chamada falhou.

### **Razão**

Tipo: `MQLONG` - saída

Se `CompCode` for `MQCC_OK`:

**MQRC\_NONE**

( 0, X'000 ') Nenhum motivo para relatar.

Se *CompCode* for MQCC\_WARNING:

**MQRC\_CHAR\_ATTRS\_TOO\_SHORT**

( 2008, X'7D8 ') Não há espaço suficiente permitido para os atributos de caracteres

**MQRC\_INT\_ATTR\_COUNT\_TOO\_SMALL**

( 2022, X'7E6 ') Não há espaço suficiente permitido para atributos de número inteiro.

**MQRC\_SELECTOR\_NOT\_FOR\_TYPE**

( 2068, X'814 ') Seletor não aplicável ao tipo de fila.

Se *CompCode* for MQCC\_FAILED:

**MQRC\_ADAPTER\_NOT\_AVAILABLE**

( 2204, X'89C ') Adaptador não disponível.

**MQRC\_ADAPTER\_SERV\_LOAD\_ERROR**

( 2130, X'852 ') Não é possível carregar o módulo de serviço do adaptador

**MQRC\_API\_EXIT\_ERROR**

( 2374, X'946 ') Saída de API com falha

**MQRC\_API\_EXIT\_LOAD\_ERROR**

( 2183, X'887 ') Não é possível carregar a saída de API

**MQRC\_ASID\_MISMATCH**

( 2157, X'86D ') Os ASIDs primário e inicial diferem.

**MQRC\_CALL\_IN\_PROGRESS**

( 2219, X'8AB ') chamada MQI inserida antes da chamada anterior ser concluída.

**MQRC\_CF\_STRUC\_FAILED**

( 2373, X'945 ') A estrutura do recurso de acoplamento falhou

**MQRC\_CF\_STRUC\_IN\_USE**

( 2346, X'92A ') Estrutura de recurso de acoplamento em uso.

**MQRC\_CHAR\_ATTR\_LENGTH\_ERROR**

( 2006, X'7D6 ') O comprimento dos atributos de caractere não é válido

**MQRC\_CHAR\_ATTRS\_ERROR**

( 2007, X'7D7 ') A cadeia de atributos de caractere não é válida.

**MQRC\_CICS\_WAIT\_FAILED**

( 2140, X'85C ') Pedido de espera rejeitado por CICS.

**MQRC\_CONNECTION\_BROKEN**

( 2009, X'7D9 ') Conexão com o gerenciador de filas perdida.

**MQRC\_CONNECTION\_NOT\_AUTHORIZED**

( 2217, X'8A9 ') Não autorizado para conexão.

**MQRC\_CONNECTION\_STOPPING**

( 2203, X'89B ') Conexão encerrando.

**MQRC\_HCONN\_ERROR**

( 2018, X'7E2 ') Identificador de conexão inválido.

**MQRC\_HOBJ\_ERROR**

( 2019, X'7E3 ') Identificador de objeto inválido.

**MQRC\_INT\_ATTR\_COUNT\_ERROR**

( 2021, X'7E5 ') A contagem de atributos inteiros não é válida.

**MQRC\_INT\_ATTRS\_ARRAY\_ERROR**

( 2023, X'7E7 ') A matriz de atributos de número inteiro não é válida.

**MQRC\_NOT\_OPEN\_FOR\_INQUIRE**

( 2038, X'7F6 ') Fila não aberta para consulta.

**MQRC\_OBJECT\_CHANGED**

(2041, X'7F9') Definição de objeto alterada desde a abertura.

**MQRC\_OBJECT\_DAMAGED**

(2101, X'835') Objeto danificado.

**MQRC\_PAGESET\_ERROR**

(2193, X'891') Erro ao acessar o conjunto de dados da página.

**MQRC\_Q\_DELETED**

(2052, X'804') Fila excluída.

**MQRC\_Q\_MGR\_NAME\_ERROR**

(2058, X'80A') Nome do gerenciador de filas inválido ou desconhecido.

**MQRC\_Q\_MGR\_NOT\_AVAILABLE**

(2059, X'80B') Gerenciador de filas não disponível para conexão.

**MQRC\_Q\_MGR\_STOPPING**

(2162, X'872') O gerenciador de filas foi encerrado.

**MQRC\_RESOURCE\_PROBLEM**

(2102, X'836') Recursos insuficientes do sistema disponíveis.

**MQRC\_SELECTOR\_COUNT\_ERROR**

(2065, X'811') Contagem de seletores inválida.

**MQRC\_SELECTOR\_ERROR**

(2067, X'813') Seletor de atributo inválido.

**MQRC\_SELECTOR\_LIMIT\_EXCEEDED**

(2066, X'812') Contagem de seletores muito grande.

**MQRC\_STORAGE\_NOT\_AVAILABLE**

(2071, X'817') Armazenamento insuficiente disponível.

**MQRC\_SUPPRESSED\_BY\_EXIT**

(2109, X'83D') Chamada suprimida pelo programa de saída.

**MQRC\_UNEXPECTED\_ERROR**

(2195, X'893') Ocorreu um erro inesperado.

Para obter informações detalhadas sobre esses códigos, consulte [Mensagens e códigos de razão](#)

## Observações de Uso

1. Os valores retornados são uma captura instantânea dos atributos selecionados. Não há garantia de que os atributos permaneçam os mesmos antes que o aplicativo possa agir nos valores retornados.
2. Ao abrir uma fila modelo, uma fila local dinâmica é criada. Uma fila local dinâmica é criada, mesmo se você abrir a fila modelo para consultar sobre seus atributos.

Os atributos da fila dinâmica são basicamente os mesmos que os atributos da fila modelo no momento em que a fila dinâmica é criada. Se você então usar a chamada MQINQ nessa fila, o gerenciador de filas retornará os atributos da fila dinâmica e não os atributos da fila modelo. Consulte [Tabela 561 na página 867](#) para obter detalhes de quais atributos da fila modelo são herdados pela filas dinâmicas.

3. Se o objeto que está sendo consultado for uma fila de alias, os valores de atributo retornados pela chamada MQINQ são os atributos da fila de alias. Não são os atributos da fila base ou do tópico para os quais o alias é resolvido.
4. Se o objeto que está sendo consultado for uma fila de clusters, os atributos que podem ser consultados dependem de como a fila é aberta:
  - É possível abrir uma fila de clusters para consultar mais uma ou mais operações de entrada, navegação ou configuração. Para isso, deve haver uma instância local da fila de clusters para a abertura ser bem-sucedida. Nesse caso, os atributos que podem ser consultados são os atributos válidos para filas locais.

Se a fila de clusters estiver aberta para consulta sem entrada, procura, ou conjunto especificado, a chamada retornará o código de conclusão MQCC\_WARNING e o código de razão MQRC\_SELECTOR\_NOT\_FOR\_TYPE (2068) se você tentar consultar os atributos que são válidos apenas para filas locais e não filas de clusters.

- É possível abrir uma fila de clusters para consulta ao transmitir o nome do gerenciador de filas base do gerenciador de filas conectado

Para isso, deve haver uma instância local da fila de clusters para a abertura ser bem-sucedida. Se o gerenciador de filas de base não for transmitido, a chamada retornará o código de conclusão MQCC\_WARNING e o código de razão MQRC\_SELECTOR\_NOT\_FOR\_TYPE (2068) se você tentar consultar atributos que são válidos apenas para filas locais e não filas de clusters

- Se a fila de clusters for aberta para consulta sozinha, ou consulta e saída, apenas os atributos listados poderão ser consultados O atributo **QType** tem o valor MQQT\_CLUSTER neste caso:

- MQCA\_Q\_DESC
- MQCA\_Q\_NAME
- MQIA\_DEF\_BIND
- MQIA\_DEF\_PERSISTENCE
- MQIA\_DEF\_PRIORITY
- MQIA\_INHIBIT\_PUT
- MQIA\_Q\_TYPE

É possível abrir a fila de clusters com nenhuma ligação fixa É possível abri-lo com o MQOO\_BIND\_NOT\_FIXED especificado na chamada MQOPEN Como alternativa, especifique MQOO\_BIND\_AS\_Q\_DEF e configure o atributo **DefBind** da fila para MQBND\_BIND\_NOT\_FIXED Se você abrir uma fila de clusters sem nenhuma ligação fixa, chamadas MQINQ sucessivas para a fila poderão consultar diferentes instâncias da fila de clusters. No entanto, é típico que todas as instâncias tenham os mesmos valores de atributos.

- Um objeto de fila de alias pode ser definido para um cluster Como TARGTYPE e TARGET não são atributos de cluster, o processo que executa um processo MQOPEN na fila de alias não reconhece o objeto para o qual o alias é resolvido.

Durante o MQOPEN inicial, a fila de alias é resolvida para um gerenciador de fila e uma fila no cluster.. A resolução do nome ocorre novamente no gerenciador de filas remotas, e é aqui que o TARGTYPE da fila de alias é resolvido

Se a fila de alias for resolvida para um alias de tópico, então a publicação de mensagens colocadas na fila de alias ocorrerá neste gerenciador de filas remotas

Consulte [Filas do cluster](#) ..

5. Você pode desejar consultar vários atributos e, em seguida, configurar alguns deles usando a chamada MQSET . Para programar a consulta e configurar com eficiência, posicione os atributos a serem configurados no início das matrizes do seletor. Se você fizer isso, as mesmas matrizes com contagem reduzida poderão ser usadas para MQSET.
6. Se mais de uma das situações de aviso surgir (consulte o parâmetro **CompCode** ), o código de razão retornado será o primeiro na lista a seguir que se aplica:
  - a. MQRC\_SELECTOR\_NOT\_FOR\_TYPE
  - b. MQRC\_INT\_ATTR\_COUNT\_TOO\_SMALL
  - c. MQRC\_CHAR\_ATTRS\_TOO\_SHORT
7. O tópico a seguir possui informações sobre atributos de objeto:
  - [“Atributos para filas” na página 865](#)
  - [“Atributos para Listas de Nomes” na página 899](#)
  - [“Atributos para Definições de Processo” na página 901](#)
  - [“Atributos do gerenciador de filas” na página 827](#)

## Chamada C

```
MQINQ (Hconn, Hobj, SelectorCount, Selectors, IntAttrCount, IntAttrs,  
CharAttrLength, CharAttrs, &CompCode, &Reason);
```

Declare os parâmetros conforme a seguir:

```
MQHCONN  Hconn;           /* Connection handle */  
MQHOBJ   Hobj;           /* Object handle */  
MQLONG   SelectorCount;  /* Count of selectors */  
MQLONG   Selectors[n];   /* Array of attribute selectors */  
MQLONG   IntAttrCount;   /* Count of integer attributes */  
MQLONG   IntAttrs[n];    /* Array of integer attributes */  
MQLONG   CharAttrLength; /* Length of character attributes buffer */  
MQCHAR   CharAttrs[n];   /* Character attributes */  
MQLONG   CompCode;       /* Completion code */  
MQLONG   Reason;        /* Reason code qualifying CompCode */
```

## Chamada COBOL

```
CALL 'MQINQ' USING HCONN, HOBJ, SELECTORCOUNT, SELECTORS-TABLE,  
INTATTRCOUNT, INTATTRS-TABLE, CHARATTRLENGTH,  
CHARATTRS, COMPCODE, REASON.
```

Declare os parâmetros conforme a seguir:

```
** Connection handle  
01 HCONN          PIC S9(9) BINARY.  
** Object handle  
01 HOBJ          PIC S9(9) BINARY.  
** Count of selectors  
01 SELECTORCOUNT PIC S9(9) BINARY.  
** Array of attribute selectors  
01 SELECTORS-TABLE.  
02 SELECTORS     PIC S9(9) BINARY OCCURS n TIMES.  
** Count of integer attributes  
01 INTATTRCOUNT PIC S9(9) BINARY.  
** Array of integer attributes  
01 INTATTRS-TABLE.  
02 INTATTRS     PIC S9(9) BINARY OCCURS n TIMES.  
** Length of character attributes buffer  
01 CHARATTRLENGTH PIC S9(9) BINARY.  
** Character attributes  
01 CHARATTRS     PIC X(n).  
** Completion code  
01 COMPCODE      PIC S9(9) BINARY.  
** Reason code qualifying COMPCODE  
01 REASON        PIC S9(9) BINARY.
```

## Chamada PL/I

```
call MQINQ (Hconn, Hobj, SelectorCount, Selectors, IntAttrCount,  
IntAttrs, CharAttrLength, CharAttrs, CompCode, Reason);
```

Declare os parâmetros conforme a seguir:

```
dcl Hconn          fixed bin(31); /* Connection handle */  
dcl Hobj          fixed bin(31); /* Object handle */  
dcl SelectorCount fixed bin(31); /* Count of selectors */  
dcl Selectors(n)  fixed bin(31); /* Array of attribute selectors */  
dcl IntAttrCount  fixed bin(31); /* Count of integer attributes */  
dcl IntAttrs(n)   fixed bin(31); /* Array of integer attributes */  
dcl CharAttrLength fixed bin(31); /* Length of character attributes  
buffer */  
dcl CharAttrs     char(n);       /* Character attributes */  
dcl CompCode      fixed bin(31); /* Completion code */
```



```
dcl Reason          fixed bin(31); /* Reason code qualifying
                                   CompCode */
```

## Chamada do Assembler de Alto Nível

```
CALL MQINQ,(HCONN,HOBJ,SELECTORCOUNT,SELECTORS,INTATTRCOUNT, X
           INTATTRS,CHARATTRLENGTH,CHARATTRS,COMP CODE,REASON)
```

Declare os parâmetros conforme a seguir:

HCONN	DS	F	Connection handle
HOBJ	DS	F	Object handle
SELECTORCOUNT	DS	F	Count of selectors
SELECTORS	DS	(n)F	Array of attribute selectors
INTATTRCOUNT	DS	F	Count of integer attributes
INTATTRS	DS	(n)F	Array of integer attributes
CHARATTRLENGTH	DS	F	Length of character attributes buffer
CHARATTRS	DS	CL(n)	Character attributes
COMP CODE	DS	F	Completion code
REASON	DS	F	Reason code qualifying COMP CODE

## Chamada do Visual Basic

```
MQINQ Hconn, Hobj, SelectorCount, Selectors, IntAttrCount, IntAttrs,
      CharAttrLength, CharAttrs, CompCode, Reason
```

Declare os parâmetros conforme a seguir:

Dim Hconn	As Long	'Connection handle'
Dim Hobj	As Long	'Object handle'
Dim SelectorCount	As Long	'Count of selectors'
Dim Selectors	As Long	'Array of attribute selectors'
Dim IntAttrCount	As Long	'Count of integer attributes'
Dim IntAttrs	As Long	'Array of integer attributes'
Dim CharAttrLength	As Long	'Length of character attributes buffer'
Dim CharAttrs	As String	'Character attributes'
Dim CompCode	As Long	'Completion code'
Dim Reason	As Long	'Reason code qualifying CompCode'

## MQINQMP-Propriedade da mensagem de consulta

A chamada MQINQMP retorna o valor de uma propriedade de uma mensagem.

### Sintaxe

MQINQMP (*Hconn, Hmsg, InqPropInqProp, Nome, PropDesc, Tipo, ValueLength, Valor, DataLength, CompCode, Motivo*)

### Parâmetros

#### Hconn

Tipo: MQHCONN - entrada

Esta manipulação representa a conexão ao gerenciador de filas. O valor de *Hconn* deve corresponder à manipulação de conexões que foi usada para criar a manipulação de mensagens especificada no parâmetro **Hmsg** ..

Se o identificador de mensagens foi criado usando MQHC\_UNASSOCIATED\_HCONN, uma conexão válida deverá ser estabelecida no encadeamento consultando uma propriedade do identificador de mensagens, caso contrário, a chamada falhará com MQRC\_CONNECTION\_BROKEN

## Hmsg

Tipo: MQHMSG-entrada

Esta é a manipulação de mensagem a ser consultada O valor foi retornado por uma chamada anterior de **MQCRTMH**

## InqPropOpções

Tipo: MQIMPO-entrada/saída

Consulte o tipo de dados [MQIMPO](#) para obter detalhes..

## Nome

Tipo: MQCHARV-entrada/saída

O nome da propriedade para consultar.

Se não for possível localizar nenhuma propriedade com esse nome, a chamada falhará com a razão **MQRC\_PROPERTY\_NOT\_AVAILABLE**

É possível usar o sinal de porcentagem do caractere curinga (%) no final do nome da propriedade.. O curinga corresponde a zero ou mais caracteres, incluindo o caractere ponto (.). Isso permite que um aplicativo inquirir o valor de muitas propriedades. Chame MQINQMP com a opção **MQIMPO\_INQ\_FIRST** para obter a primeira propriedade correspondente e novamente com a opção **MQIMPO\_INQ\_NEXT** para obter a próxima propriedade correspondente. Quando mais nenhuma propriedade correspondente estiver disponível, a chamada falhará com **MQRC\_PROPERTY\_NOT\_AVAILABLE** Se o campo *ReturnedName* da estrutura InqPropOpts for inicializado com um endereço ou deslocamento para o nome retornado da propriedade, isso será concluído no retorno de MQINQMP com o nome da propriedade correspondente. Se o campo *VSBufSize* do *ReturnedName* na estrutura de Opts InqPropfor menor que o comprimento do nome da propriedade retornado, o código de conclusão será configurado **MQCC\_FAILED** com a razão **MQRC\_PROPERTY\_NAME\_TOO\_BIG**.

Propriedades que possuem sinônimos conhecidos são retornadas da seguinte forma:

1. Propriedades com o prefixo "mqps." são retornados como o nome da propriedade IBM MQ .. Por exemplo, "MQTopicString" é o nome retornado em vez de "mqps.Top"
2. Propriedades com o prefixo "jms." ou "McD". são retornados como o nome do campo de cabeçalho JMS , por exemplo, "JMSExpiration" é o nome retornado em vez de "jms.Exp". "
3. Propriedades com o prefixo "usr." são retornados sem esse prefixo, por exemplo, "Color" é retornado em vez de "usr.Color".

As propriedades com sinônimos são retornadas apenas uma vez

Na linguagem de programação C, as seguintes variáveis macro são definidas para consultar todas as propriedades e, em seguida, todas as propriedades que começam com "usr.":

### **MQPROP\_INQUIRE\_ALL**

Consultar em todas as propriedades da mensagem

MQPROP\_INQUIRE\_ALL pode ser usado da seguinte maneira:

```
MQCHARV Name = {MQPROP_INQUIRE_ALL};
```

### **MQPROP\_INQUIRE\_ALL\_USR**

Consulte todas as propriedades da mensagem que iniciam "usr.". O nome retornado é retornado sem o "usr." .

Se MQIMP\_INQ\_NEXT for especificado, mas o Nome tiver mudado desde a chamada anterior ou essa for a primeira chamada, então MQIMPO\_INQ\_FIRST será implícito.

Consulte [Nomes de Propriedade e Restrições de Nome de Propriedade](#) para obter informações adicionais sobre o uso de nomes de propriedade

## PropDesc

Tipo: MQPD-saída

Essa estrutura é utilizada para definir os atributos de uma propriedade, incluindo o que acontece se a propriedade não for suportada, a qual contexto de mensagem a propriedade pertence e para quais mensagens a propriedade deve ser copiada. Consulte [MQPD](#) para obter detalhes dessa estrutura..

#### tipo

Tipo: MQLONG-entrada/saída

No retorno da chamada MQINQMP, esse parâmetro é configurado para o tipo de dados de *Valor*. O tipo de dados pode ser qualquer um dos seguintes:

#### **MQTYPE\_BOOLEAN**

Um booleano.

#### **MQTYPE\_BYTE\_STRING**

Uma cadeia de bytes

#### **MQTYPE\_INT8**

Um número inteiro assinado de 8 bits.

#### **MQTYPE\_INT16**

Um número inteiro assinado de 16 bits.

#### **MQTYPE\_INT32**

Um número inteiro assinado de 32 bits.

#### **MQTYPE\_INT64**

Um número inteiro assinado de 64 bits.

#### **MQTYPE\_FLOAT32**

Um número de vírgula flutuante de 32 bits..

#### **MQTYPE\_FLOAT64**

Um número de vírgula flutuante de 64 bits..

#### **MQTYPE\_STRING**

Uma sequência de Caracteres

#### **MQTYPE\_NULL**

A propriedade existe mas possui um valor nulo.

Se o tipo de dados do valor da propriedade não for reconhecido, MQTYPE\_STRING será retornado e uma representação em sequência do valor será colocada na área *Valor*. Uma representação em sequência do tipo de dados pode ser localizada no campo *TypeString* no parâmetro *InqPropOpts* .. Um código de conclusão de aviso é retornado com a razão MQRC\_PROP\_TYPE\_NOT\_SUPPORTED.

Além disso, se a opção MQIMPO\_CONVERT\_TYPE for especificada, será solicitada a conversão do valor da propriedade. Use *Tipo* como uma entrada para especificar o tipo de dados como o qual você deseja que a propriedade seja retornada. Consulte a descrição da opção [MQIMPO\\_CONVERT\\_TYPE](#) da estrutura [MQIMPO](#) para obter detalhes de conversão de tipo de dados

Se você não solicitar a conversão de tipo, poderá usar o seguinte valor na entrada:

#### **MQTYPE\_AS\_SET**

O valor da propriedade é retornado sem converter seu tipo de dados.

#### **ValueLength**

Tipo: MQLONG - entrada

O comprimento em bytes da área Valor. Especifique zero para propriedades para as quais não é necessário o valor retornado. Elas podem ser propriedades projetadas por um aplicativo para ter um valor nulo ou uma sequência vazia. Especifique também zero se a opção [MQIMPO\\_QUERY\\_LENGTH](#) tiver sido especificada; nesse caso, nenhum valor será retornado.

#### **Value**

Tipo: MQBYTEX *ValueLength* -saída

Essa é a área para conter o valor da propriedade consultada. O buffer deve ser alinhado em um limite apropriado para o valor retornado. Não fazer isso pode resultar em um erro quando o valor for acessado posteriormente.

Se *ValueLength* for menor que o comprimento do valor da propriedade, o máximo do valor da propriedade possível será movido para *Value* e a chamada falhará com o código de conclusão MQCC\_FAILED e a razão MQRC\_PROPERTY\_BIG.

O conjunto de caracteres dos dados no *Valor* é fornecido pelo campo ReturnedCCSID no parâmetro Opts InqProp. A codificação dos dados em *Valor* é fornecida pelo campo ReturnedEncoding no parâmetro Opts InqProp

Na linguagem de programação C, o parâmetro é declarado como um ponteiro para vazio; o endereço de qualquer tipo de dados pode ser especificado como o parâmetro.

Se o parâmetro *ValueLength* for zero, *Value* não será referido e seu valor transmitido por programas gravados no assembler C ou System/390 poderá ser nulo.

### **DataLength**

Tipo: MQLONG - saída

Esse é o comprimento em bytes do valor da propriedade real, conforme retornado na área *Valor* .

Se *DataLength* for menor que o comprimento do valor da propriedade, *DataLength* ainda será preenchido no retorno da chamada MQINQMP. Isso permite que o aplicativo determine o tamanho do buffer necessário para acomodar o valor da propriedade e, em seguida, emita novamente a chamada com um buffer do tamanho apropriado..

Os valores a seguir também podem ser retornadas:

Se o parâmetro *Type* estiver configurado como MQTYPE\_STRING ou MQTYPE\_BYTE\_STRING:

#### **MQVL\_EMPTY\_STRING**

A propriedade existe mas não contém caracteres ou bytes.

### **CompCode**

Tipo: MQLONG - saída

O código de conclusão; é um dos seguintes:

#### **MQCC\_OK**

Indica conclusão bem-sucedida.

#### **MQCC\_WARNING**

Aviso (conclusão parcial).

#### **MQCC\_FAILED**

A chamada falhou.

### **Razão**

Tipo: MQLONG - saída

Se *CompCode* for MQCC\_OK:

#### **MQRC\_NONE**

(0, X'000') Nenhuma razão a relatar.

Se *CompCode* for MQCC\_WARNING:

#### **MQRC\_PROP\_NAME\_NOT\_CONVERTED**

(2492, X'09BC') Nome da propriedade retornada não convertido.

#### **MQRC\_PROP\_VALUE\_NOT\_CONVERTED**

(2466, X'09A2') Valor da propriedade não convertido.

#### **MQRC\_PROP\_TYPE\_NOT\_SUPPORTED**

(2467, X'09A3') O tipo de dados da propriedade não é suportado

#### **MQRC\_RFH\_FORMAT\_ERROR**

(2421, X'0975 ') Uma pasta MQRFH2 contendo propriedades não pôde ser analisada.

Se *CompCode* for MQCC\_FAILED:

#### **MQRC\_ADAPTER\_NOT\_AVAILABLE**

(2204, X'089C') Adaptador não disponível.

**MQR\_ADAPTER\_SERV\_LOAD\_ERROR**

(2130, X'0852 ') Impossível carregar o módulo de serviço do adaptador.

**MQR\_ASID\_MISMATCH**

(2157, X'086D') Os ASIDs primário e inicial diferem.

**MQR\_BUFFER\_ERROR**

(2004, X'07D4') Parâmetro de valor inválido.

**MQR\_BUFFER\_LENGTH\_ERROR**

(2005, X'07D5') Parâmetro de comprimento do valor inválido.

**MQR\_CALL\_IN\_PROGRESS**

(2219, X'08AB') chamada MQI inserida antes da chamada anterior ser concluída.

**MQR\_CONNECTION\_BROKEN**

(2009, X'07D9') Conexão com o gerenciador de filas perdida.

**MQR\_DATA\_LENGTH\_ERROR**

(2010, X'07DA') Parâmetro de comprimento de dados inválido.

**MQR\_IMPO\_ERROR**

(2464, X'09A0') A estrutura de opções da propriedade de mensagem não é válida.

**MQR\_HMSG\_ERROR**

(2460, X'099C') Identificador de mensagem inválido.

**MQR\_MSG\_HANDLE\_IN\_USE**

(2499, X'09C3') Identificador de mensagem já em uso.

**MQR\_OPTIONS\_ERROR**

(2046, X'07F8') Opções não válidas ou não consistentes.

**MQR\_PD\_ERROR**

(2482, X'09B2') Estrutura do descritor de propriedade inválida.

**MQR\_PROP\_CONV\_NOT\_SUPPORTED**

(2470, X'09A6') Conversão do tipo de dados real para solicitado não suportada.

**MQR\_PROPERTY\_NAME\_ERROR**

(2442, X'098A') Nome da propriedade inválido

**MQR\_PROPERTY\_NAME\_TOO\_BIG**

(2465, X'09A1') Nome da propriedade muito grande para o buffer de nome retornado.

**MQR\_PROPERTY\_NOT\_AVAILABLE**

(2471, X'09A7) Propriedade não disponível.

**MQR\_PROPERTY\_VALUE\_TOO\_BIG**

(2469, X'09A5') Valor da propriedade muito grande para a área Valor.

**MQR\_PROP\_NUMBER\_FORMAT\_ERROR**

(2472, X'09A8') Erro de formato numérico encontrado nos dados de valor...

**MQR\_PROPERTY\_TYPE\_ERROR**

(2473, X'09A9') Tipo de propriedade solicitado inválido..

**MQR\_SOURCE\_CCSID\_ERROR**

(2111, X'083F') Identificador do conjunto de caracteres codificados do nome da propriedade inválido.

**MQR\_STORAGE\_NOT\_AVAILABLE**

(2071, X'0871 ') Armazenamento insuficiente disponível.

**MQR\_UNEXPECTED\_ERROR**

(2195, X'0893 ') Ocorreu um erro inesperado.

Para obter informações detalhadas sobre esses códigos, consulte [Mensagens e códigos de razão](#).

## Chamada C

```
MQINQMP (Hconn, Hmsg, &InqPropOpts, &Name, &PropDesc, &Type,  
ValueLength, Value, &DataLength, &CompCode, &Reason);
```

Declare os parâmetros conforme a seguir:

```
MQHCONN Hconn; /* Connection handle */  
MQHMSG Hmsg; /* Message handle */  
MQIMPO InqPropOpts; /* Options that control the action of MQINQMP */  
MQCHARV Name; /* Property name */  
MQPD PropDesc; /* Property descriptor */  
MQLONG Type; /* Property data type */  
MQLONG ValueLength; /* Length in bytes of the Value area */  
MQBYTE Value[n]; /* Area to contain the property value */  
MQLONG DataLength; /* Length of the property value */  
MQLONG CompCode; /* Completion code */  
MQLONG Reason; /* Reason code qualifying CompCode */
```

## Chamada COBOL

```
CALL 'MQINQMP' USING HCONN, HMSG, INQMSGOPTS, NAME, PROPDESC, TYPE,  
VALUELENGTH, VALUE, DATALENGTH, COMPCODE, REASON.
```

Declare os parâmetros conforme a seguir:

```
** Connection handle  
01 HCONN PIC S9(9) BINARY.  
** Message handle  
01 HMSG PIC S9(18) BINARY.  
** Options that control the action of MQINQMP  
01 INQMSGOPTS.  
COPY CMQIMPOV.  
** Property name  
01 NAME.  
COPY CMQCHRVV.  
** Property descriptor  
01 PROPDESC.  
COPY CMQPDV.  
** Property data type  
01 TYPE PIC S9(9) BINARY.  
** Length in bytes of the VALUE area  
01 VALUELENGTH PIC S9(9) BINARY.  
** Area to contain the property value  
01 VALUE PIC X(n).  
** Length of the property value  
01 DATALENGTH PIC S9(9) BINARY.  
** Completion code  
01 COMPCODE PIC S9(9) BINARY.  
** Reason code qualifying COMPCODE  
01 REASON PIC S9(9) BINARY.
```

## Chamada PL/I

```
call MQINQMP (Hconn, Hmsg, InqPropOpts, Name, PropDesc, Type,  
ValueLength, Value, DataLength, CompCode, Reason);
```

Declare os parâmetros conforme a seguir:

```
dc1 Hconn fixed bin(31); /* Connection handle */  
dc1 Hmsg fixed bin(63); /* Message handle */  
dc1 InqPropOpts like MQIMPO; /* Options that control the action of MQINQMP */  
dc1 Name like MQCHARV; /* Property name */  
dc1 PropDesc like MQPD; /* Property descriptor */  
dc1 Type fixed bin (31); /* Property data type */  
dc1 ValueLength fixed bin (31); /* Length in bytes of the Value area */
```

```

dcl Value          char (n);          /* Area to contain the property value */
dcl DataLength    fixed bin (31);    /* Length of the property value */
dcl CompCode      fixed bin (31);    /* Completion code */
dcl Reason        fixed bin (31);    /* Reason code qualifying CompCode */

```

## Chamada do Assembler de Alto Nível

```

CALL MQINQMP, (HCONN, HMSG, INQMSGOPTS, NAME, PROPDESC, TYPE,
VALUELENGTH, VALUE, DATALENGTH, COMPCODE, REASON)

```

Declare os parâmetros conforme a seguir:

HCONN	DS	F	Connection handle
HMSG	DS	D	Message handle
INQMSGOPTS	CMQIMPOA	,	Options that control the action of MQINQMP
NAME	CMQCHRVA	,	Property name
PROPDESC	CMQPDA	,	Property descriptor
TYPE	DS	F	Property data type
VALUELENGTH	DS	F	Length in bytes of the VALUE area
VALUE	DS	CL(n)	Area to contain the property value
DATALENGTH	DS	F	Length of the property value
COMPCODE	DS	F	Completion code
REASON	DS	F	Reason code qualifying COMPCODE

## MQMHBUF-Converter identificador de mensagem em buffer

A chamada MQMHBUF converte um identificador de mensagem em um buffer e é o inverso da chamada MQBUFMH

### Sintaxe

MQMHBUF (*Hconn*, *Hmsg*, *MsgHBufOpts*, *Name*, *MsgDesce BufferLength*, *Buffer*, *DataLength*, *CompCode*, *Reason*)

### Parâmetros

#### Hconn

Tipo: MQHCONN - entrada

Esta manipulação representa a conexão ao gerenciador de filas. O valor de *Hconn* deve corresponder à manipulação de conexões que foi usada para criar a manipulação de mensagens especificada no parâmetro **Hmsg** ..

Se a manipulação de mensagens foi criada usando MQHC\_UNASSOCIATED\_HCONN, uma conexão válida deverá ser estabelecida no encadeamento excluindo a manipulação de mensagens. Se uma conexão válida não for estabelecida a chamada falha com MQRC\_CONNECTION\_BROKEN.

#### Hmsg

Tipo: MQHMSG-entrada

Este é o identificador de mensagem para o qual um buffer é necessário. O valor foi retornado por uma chamada MQCRTMH anterior.

#### MsgHBufOpts

Tipo: MQMHBO-entrada

A estrutura MQMHBO permite que os aplicativos especifiquem opções que controlam como os buffers são produzidos a partir de manipulações de mensagens.

Consulte [“MQMHBO-Identificador de mensagens para opções de buffer”](#) na página 495 para obter detalhes.

#### Nome

Tipo: MQCHARV-entrada

O nome da propriedade ou propriedades a serem colocadas no buffer.

Se nenhuma propriedade correspondente ao nome puder ser localizada, a chamada falha com MQRC\_PROPERTY\_NOT\_AVAILABLE.

É possível usar um curinga para colocar mais de uma propriedade no buffer. Para isso, use o caractere curinga '%' no final do nome da propriedade. Esse curinga corresponde a zero ou mais caracteres, incluindo o '!':

Na linguagem de programação C, as variáveis de macro a seguir são definidas para consultar todas as propriedades e todas as propriedades que começam com 'usr':

#### **MQPROP\_INQUIRE\_ALL**

Colocar todas as propriedades da mensagem no buffer

#### **MQPROP\_INQUIRE\_ALL\_USR**

Coloque todas as propriedades da mensagem que começam com os caracteres 'usr.' no buffer.

Consulte [Nomes de Propriedade e Restrições de Nome de Propriedade](#) para obter informações adicionais sobre o uso de nomes de propriedade

### **MsgDesc**

Tipo: MQMD - entrada/saída

A estrutura *MsgDesc* descreve o conteúdo da Área de Buffer

Na saída, os campos *Encoding*, *CodedCharSetId* e *Format* são configurados para descrever corretamente a codificação, o identificador do conjunto de caracteres e o formato dos dados na área de buffer, conforme gravados pela chamada

Os dados nessa estrutura estão no conjunto de caracteres e na codificação do aplicativo

### **BufferLength**

Tipo: MQLONG - entrada

*BufferLength* é o comprimento da área Buffer, em bytes.

### **de Saída**

Tipo: MQBYTEXBufferLength - saída

*Buffer* define a área para conter as propriedades de mensagem.. Deve-se alinhar o buffer em um limite de 4 bytes

Se *BufferLength* for menor que o comprimento necessário para armazenar as propriedades em *Buffer*, o MQMHBUFF falhará com MQRC\_PROPERTY\_VALUE\_TOO\_BIG

O conteúdo do buffer pode ser alterado mesmo se a chamada falhar.

### **DataLength**

Tipo: MQLONG - saída

*DataLength* é o comprimento, em bytes, das propriedades retornadas no buffer.. Se o valor for zero, nenhuma propriedade correspondeu ao valor fornecido em *Name* e a chamada falha com o código de razão MQRC\_PROPERTY\_NOT\_AVAILABLE.

Se *BufferLength* for menor que o comprimento necessário para armazenar as propriedades no buffer, a chamada MQMHBUFF falhará com MQRC\_PROPERTY\_VALUE\_TOO\_BIG, mas um valor ainda será inserido em *DataLength*. Isso permite que o aplicativo determine o tamanho do buffer necessário para acomodar as propriedades e, em seguida, emita novamente a chamada com o *BufferLength* necessário

### **CompCode**

Tipo: MQLONG - saída

O código de conclusão; é um dos seguintes:

#### **MQCC\_OK**

Indica conclusão bem-sucedida.



**MQCC\_FAILED**

A chamada falhou.

**Razão**

Tipo: MQLONG - saída

O código de razão que qualifica *CompCode*.

Se *CompCode* for MQCC\_OK:

**MQRC\_NONE**

(0, X'000') Nenhuma razão a relatar.

Se *CompCode* for MQCC\_FAILED:

**MQRC\_ADAPTER\_NOT\_AVAILABLE**

(2204, X'089C') Adaptador não disponível.

**MQRC\_ADAPTER\_SERV\_LOAD\_ERROR**

(2130, X'852') Não foi possível carregar o módulo de serviço do adaptador.

**MQRC\_ASID\_MISMATCH**

(2157, X'86D') ASIDs de início e primárias são diferentes.

**MQRC\_MHBO\_ERROR**

(2501, X'095C') O identificador de mensagens para a estrutura de opções de buffer não é válido..

**MQRC\_BUFFER\_ERROR**

(2004, X'07D4') Parâmetro de buffer inválido.

**MQRC\_BUFFER\_LENGTH\_ERROR**

(2005, X'07D5') Parâmetro de comprimento do buffer inválido.

**MQRC\_CALL\_IN\_PROGRESS**

(2219, X'08AB') chamada MQI inserida antes da chamada anterior ser concluída.

**MQRC\_CONNECTION\_BROKEN**

(2009, X'07D9') Conexão com o gerenciador de filas perdida.

**MQRC\_DATA\_LENGTH\_ERROR**

(2010, X'07DA') Parâmetro de comprimento de dados inválido.

**MQRC\_HMSG\_ERROR**

(2460, X'099C') Identificador de mensagem inválido.

**MQRC\_MD\_ERROR**

(2026, X'07EA') Descritor de mensagens inválido.

**MQRC\_MSG\_HANDLE\_IN\_USE**

(2499, X'09C3') Identificador de mensagem já em uso.

**MQRC\_OPTIONS\_ERROR**

(2046, X'07FE') Opções não válidas ou não consistentes.

**MQRC\_PROPERTY\_NAME\_ERROR**

(2442, X'098A') O nome da propriedade não é válido.

**MQRC\_PROPERTY\_NOT\_AVAILABLE**

(2471, X'09A7') Propriedade não disponível.

**MQRC\_PROPERTY\_VALUE\_TOO\_BIG**

(2469, X'09A5') BufferLength valor é muito pequeno para conter propriedades especificadas.

**MQRC\_UNEXPECTED\_ERROR**

(2195, X'893') Ocorreu um erro inesperado.

Para obter informações detalhadas sobre esses códigos, consulte [Mensagens e códigos de razão](#).

## Chamada C

```
MQMHBUF (Hconn, Hmsg, &MsgHBufOpts, &Name, &MsgDesc, BufferLength, Buffer,  
&DataLength, &CompCode, &Reason);
```

Declare os parâmetros conforme a seguir:

```
MQHCONN Hconn;          /* Connection handle */  
MQHMSG Hmsg;            /* Message handle */  
MQMHBO MsgHBufOpts;    /* Options that control the action of MQMHBUF */  
MQCHARV Name;          /* Property name */  
MQMD MsgDesc;           /* Message descriptor */  
MQLONG BufferLength;    /* Length in bytes of the Buffer area */  
MQBYTE Buffer[n];       /* Area to contain the properties */  
MQLONG DataLength;     /* Length of the properties */  
MQLONG CompCode;       /* Completion code */  
MQLONG Reason;         /* Reason code qualifying CompCode */
```

## Observações de Uso

O MQMHBUF converte um identificador de mensagem em um buffer

É possível usá-lo com uma saída de API MQGET para acessar determinadas propriedades, usando as APIs de propriedade de mensagens e, em seguida passado-as em um buffer de volta para um aplicativo projetado para usar cabeçalhos MQRFH2 em vez de identificadores de mensagens.

Essa chamada é o inverso da chamada MQBUFMH, que pode ser usada para analisar as propriedades de mensagem de um buffer em uma manipulação de mensagens

## Chamada COBOL

```
CALL 'MQMHBUF' USING HCONN, HMSG, MSGHBUFOPTS, NAME, MSGDESC,  
BUFFERLENGTH, BUFFER, DATALENGTH, COMPCODE, REASON.
```

Declare os parâmetros conforme a seguir:

```
** Connection handle  
01 HCONN          PIC S9(9) BINARY.  
** Message handle  
01 HMSG          PIC S9(18) BINARY.  
** Options that control the action of MQMHBUF  
01 MSGHBUFOPTS.  
   COPY CMQMHBV.  
** Property name  
01 NAME  
   COPY CMQCHRVV.  
** Message descriptor  
01 MSGDESC  
   COPY CMQMDV.  
** Length in bytes of the Buffer area */  
01 BUFFERLENGTH PIC S9(9) BINARY.  
** Area to contain the properties  
01 BUFFER        PIC X(n).  
** Length of the properties  
01 DATALENGTH  PIC S9(9) BINARY.  
** Completion code  
01 COMPCODE     PIC S9(9) BINARY.  
** Reason code qualifying COMPCODE  
01 REASON       PIC S9(9) BINARY.
```

## Chamada PL/I

```
call MQMHBUF (Hconn, Hmsg, MsgHBufOpts, Name, MsgDesc, BufferLength, Buffer,  
DataLength, CompCode, Reason);
```

Declare os parâmetros conforme a seguir:

```
dcl Hconn          fixed bin(31); /* Connection handle */
dcl Hmsg           fixed bin(63); /* Message handle */
dcl MsgHBufOpts   like MQMHBO; /* Options that control the action of MQMHBUF */
dcl Name          like MQCHARV; /* Property name */
dcl MsgDesc       like MQMD; /* Message descriptor */
dcl BufferLength   fixed bin(31); /* Length in bytes of the Buffer area */
dcl Buffer         char(n); /* Area to contain the properties */
dcl DataLength    fixed bin(31); /* Length of the properties */
dcl CompCode      fixed bin(31); /* Completion code */
dcl Reason        fixed bin(31); /* Reason code qualifying CompCode */
```

## Chamada do Assembler de Alto Nível

```
CALL MQMHBUF, (HCONN, HMSG, MSGHBUFOPTS, NAME, MSGDESC, BUFFERLENGTH,
              BUFFER, DATALENGTH, COMPCODE, REASON)
```

Declare os parâmetros conforme a seguir:

HCONN	DS	F	Connection handle
HMSG	DS	D	Message handle
MSGHBUFOPTS	CMQMHBOA	,	Options that control the action of MQMHBUF
NAME	CMQCHRVA	,	Property name
MSGDESC	CMQMDA	,	Message descriptor
BUFFERLENGTH	DS	F	Length in bytes of the BUFFER area
BUFFER	DS	CL(n)	Area to contain the properties
DATALENGTH	DS	F	Length of the properties
COMPCODE	DS	F	Completion code
REASON	DS	F	Reason code qualifying COMPCODE

## MQOPEN-Abrir objeto

A chamada MQOPEN estabelece acesso a um objeto.

Os seguintes tipos de objeto são válidos:

- Fila (incluindo listas de distribuição)
- Lista de Nomes
- Definição de processo
- Gerenciador de filas
- Tópico

## Sintaxe


MQOPEN (*Hconn*, *ObjDesc*, *Opções*, *Hobj*, *CompCode*, *Motivo*)

## Parâmetros

### Hconn

Tipo: MQHCONN - entrada

Esta manipulação representa a conexão ao gerenciador de filas. O valor de Hconn foi retornado por uma chamada anterior MQCONN ou MQCONNX.

 Em z/OS para aplicativos CICS, a chamada MQCONN pode ser omitida e o seguinte valor especificado para *Hconn*:

### MQHC\_DEF\_HCONN

Manipulação de conexões padrão.

## ObjDesc

Tipo: MQOD-entrada / saída

Esta é uma estrutura que identifica o objeto a ser aberto; consulte [“MQOD-Descrição de objetos”](#) na página 497 para obter detalhes

Se o campo `ObjectName` no parâmetro **ObjDesc** for o nome de uma fila modelo, uma fila local dinâmica é criada com os atributos da fila modelo; isso acontece independentemente das opções especificadas no parâmetro **Options**. Operações subsequentes usando o `Hobj` retornado pela chamada `MQOPEN` são executadas na nova fila dinâmica e não na fila modelo. Isso é verdadeiro até mesmo para as chamadas `MQINQ` e `MQSET`. O nome da fila modelo no parâmetro **ObjDesc** é substituído pelo nome da fila dinâmica criada.. O tipo da fila dinâmica é determinado pelo valor do atributo **DefinitionType** da fila modelo (consulte [“Atributos para filas”](#) na página 865).. Para obter informações sobre as opções de fechamento aplicáveis a filas dinâmicas, consulte a descrição da chamada `MQCLOSE`

## Opções

Tipo: MQLONG - entrada

Você deve especificar pelo menos uma das opções a seguir:

- `MQOO_BROWSE`
- `MQOO_INPUT_*` (apenas um deles)..
- `MQOO_INQUIRE`
- `MQOO_OUTPUT`
- `MQOO_SET`
- `MQOO_BIND_*` (apenas um deles)..

Consulte a tabela a seguir para obter detalhes sobre essas opções; outras opções podem ser especificadas conforme necessário Para especificar mais de uma opção, inclua os valores juntos (não inclua a mesma constante mais de uma vez) ou combine os valores usando a operação OR bit a bit (se a linguagem de programação suportar operações de bit). As combinações que não são válidas são anotadas; todas as outras combinações são válidas Apenas opções aplicáveis ao tipo de objeto especificado por `ObjDesc` são permitidas.

Tabela 553. Opções `MQOPEN` válidas para filas e tópicos

Opção	Alias <sup>1</sup>	Local e Modelo	Remoto	Cluster não local	Lista de distribuição	Tópico
<code>MQOO_INPUT_AS_Q_DEF</code>	Sim	Sim	No	No	No	No
<code>MQOO_INPUT_SHARED</code>	Sim	Sim	No	No	No	No
<code>MQOO_INPUT_EXCLUSIVE</code>	Sim	Sim	No	No	No	No
<code>MQOO_OUTPUT</code>	Sim	Sim	Sim	Sim	Sim	Sim
<code>MQOO_BROWSE</code>	Sim	Sim	No	No	No	No
<code>MQOO_CO_OP</code>	Sim	Sim	No	No	No	No
<code>MQOO_INQUIRE</code>	Sim	Sim	<u>2</u>	Sim	No	No
<code>MQOO_SET</code>	Sim	Sim	<u>2</u>	No	No	No
<code>MQOO_BIND_ON_OPEN</code> <sup>3</sup>	Sim	Sim	Sim	Sim	Sim	No
<code>MQOO_BIND_NOT_FIXED</code> <sup>3</sup>	Sim	Sim	Sim	Sim	Sim	No
<code>MQOO_BIND_ON_GROUP</code> <sup>3</sup>	Sim	Sim	Sim	Sim	Sim	No
<code>MQOO_BIND_AS_Q_DEF</code> <sup>3</sup>	Sim	Sim	Sim	Sim	Sim	No
<code>MQOO_SAVE_ALL_CONTEXT</code>	Sim	Sim	No	No	No	No
<code>MQOO_PASS_IDENTITY_CONTEXT</code>	Sim	Sim	Sim	Sim	Sim	<u>4</u>
<code>MQOO_PASS_ALL_CONTEXT</code>	Sim	Sim	Sim	Sim	Sim	Sim

Tabela 553. Opções MQOPEN válidas para filas e tópicos (continuação)

Opção	Alias <sup>1</sup>	Local e Modelo	Remoto	Cluster não local	Lista de distribuição	Tópico
<u>MQOO_SET_IDENTITY_CONTEXT</u>	Sim	Sim	Sim	Sim	Sim	<u>4</u>
<u>MQOO_SET_ALL_CONTEXT</u>	Sim	Sim	Sim	Sim	Sim	Sim
<u>MQOO_NO_READ_AHEAD</u>	Sim	Sim	No	No	No	No
<u>MQOO_READ_AHEAD</u>	Sim	Sim	No	No	No	No
<u>MQOO_READ_AHEAD_AS_Q_DEF</u>	Sim	Sim	No	No	No	No
<u>MQOO_ALTERNATE_USER_AUTHORITY</u>	Sim	Sim	Sim	Sim	Sim	Sim
<u>MQOO_FAIL_IF QUIESCING</u>	Sim	Sim	Sim	Sim	Sim	Sim
<u>MQOO_RESOLVE_LOCAL_Q</u>	Sim	Sim	Sim	Sim	No	No
<u>MQOO_RESOLVE_LOCAL_TOPIC</u>	No	No	No	No	No	Sim
<u>MQOO_NO_MULTICAST</u>	No	No	No	No	No	Sim

**Notas:**

1. A validade de opções para aliases depende da validade da opção para a fila para a qual o alias é resolvido
2. Esta opção é válida apenas para a definição local de uma fila remota
3. Esta opção pode ser especificada para qualquer tipo de fila, mas será ignorada, se a fila não for uma fila de clusters No entanto, o atributo da fila **DefBind** substitui a fila base, mesmo quando a fila de alias não está em um cluster
4. Esses atributos podem ser utilizados com um tópico, mas afetam apenas o contexto configurado para a mensagem retida, não os campos de contextos enviados para qualquer assinante

**Opções de Acesso:** As opções a seguir controlam o tipo de operações que podem ser executadas no objeto:

**MQOO\_INPUT\_AS\_Q\_DEF**

Abrir fila para obter mensagens usando padrão definido pela fila.

A fila é aberta para uso com as chamadas MQGET subsequentes O tipo de acesso é compartilhado ou exclusivo, dependendo do valor do atributo da fila **DefInputOpenOption** ; consulte “Atributos para filas” na página 865 para obter detalhes.

Esta opção é válida apenas para filas locais, de alias e de modelo; não é válida para filas remotas, listas de distribuições e objetos que não são filas.

**MQOO\_INPUT\_SHARED**

Abrir fila para obter mensagens com acesso compartilhado.

A fila é aberta para uso com as chamadas MQGET subsequentes A chamada poderá ser bem-sucedida se a fila estiver atualmente aberta por este ou outro aplicativo com MQOO\_INPUT\_SHARED, mas falhará com o código de razão MQRC\_OBJECT\_IN\_USE se a fila estiver atualmente aberta com MQOO\_INPUT\_EXCLUSIVE

Esta opção é válida apenas para filas locais, de alias e de modelo; não é válida para filas remotas, listas de distribuições e objetos que não são filas.

**MQOO\_INPUT\_EXCLUSIVE**

Abra a fila para obter mensagens com acesso exclusivo

A fila é aberta para uso com as chamadas MQGET subsequentes A chamada falha com o código de razão MQRC\_OBJECT\_IN\_USE se a fila estiver atualmente aberta por este ou outro aplicativo para entrada de qualquer tipo (MQOO\_INPUT\_SHARED ou MQOO\_INPUT\_EXCLUSIVE).

Esta opção é válida apenas para filas locais, de alias e de modelo; não é válida para filas remotas, listas de distribuições e objetos que não são filas.

## MQOO\_OUTPUT

Abra a fila para colocar mensagens, ou um tópico ou sequência de tópicos para publicar mensagens

A fila ou tópico é aberto para uso com chamadas MQPUT subsequentes.

Uma chamada MQOPEN com essa opção pode ser bem-sucedida mesmo se o atributo da fila **InhibitPut** estiver configurado como MQQA\_PUT\_INIBITED (embora chamadas MQPUT subsequentes falhem enquanto o atributo é configurado para esse valor).

Esta opção é válida para todos os tipos de fila, incluindo listas de distribuição, e tópicos

As notas a seguir se aplicam a essas opções:

- Apenas uma dessas opções pode ser especificada..
- Uma chamada MQOPEN com uma dessas opções pode ser bem-sucedida mesmo se o atributo da fila **InhibitGet** estiver configurado como MQQA\_GET\_INIBITED (embora chamadas MQGET subsequentes falhem enquanto o atributo estiver configurado para esse valor).
- Se a fila for definida como não compartilhável (ou seja, o atributo da fila **Shareability** tiver o valor MQQA\_NOT\_SHAREABLE), as tentativas de abrir a fila para acesso compartilhado serão tratadas como tentativas de abrir a fila com acesso exclusivo.
- Se uma fila de alias for aberta com uma dessas opções, o teste para uso exclusivo (ou para saber se outro aplicativo possui uso exclusivo) será contra a fila base para a qual o alias é resolvido.
- Essas opções não serão válidas se **ObjectQMgrName** for o nome de um alias do gerenciador de filas; isso será verdadeiro mesmo se o valor do atributo **RemoteQMgrName** na definição local de uma fila remota usada para alias do gerenciador de filas for o nome do gerenciador de filas locais.

## MQOO\_BROWSE

Abrir fila para procurar mensagens.

A fila é aberta para uso com as chamadas MQGET subsequentes com uma das seguintes opções:

- MQGMO\_BROWSE\_FIRST
- MQGMO\_BROWSE\_NEXT
- MQGMO\_BROWSE\_MSG\_UNDER\_CURSOR

Isso é permitido, mesmo se a fila estiver aberta atualmente para MQOO\_INPUT\_EXCLUSIVE Uma chamada MQOPEN com a opção MQOO\_BROWSE estabelece um cursor de procura e o posiciona logicamente antes da primeira mensagem na fila; consulte o [campo MQGMO-Options](#) para obter informações adicionais

Esta opção é válida apenas para filas locais, de alias e de modelo; não é válida para filas remotas, listas de distribuições e objetos que não são filas. Ele também não é válido se **ObjectQMgrName** for o nome de um alias de gerenciador de filas; isso é verdadeiro mesmo se o valor do atributo **RemoteQMgrName** na definição local de uma fila remota usada para alias de gerenciador de filas for o nome do gerenciador de filas locais.

## MQOO\_CO\_OP

Abra como um membro cooperante do conjunto de alças

Essa opção é válida apenas com a opção MQOO\_BROWSE Se for especificado sem MQOO\_BROWSE, MQOPEN retorna com MQRC\_OPTIONS\_ERROR.

O identificador retornado é considerado um membro de um conjunto de identificadores cooperantes para as chamadas MQGET subsequentes com uma das seguintes opções:

- MQGMO\_MARK\_BROWSE\_CO\_OP
- MQGMO\_UNMARKED\_BROWSE\_MSG
- MQGMO\_UNMARK\_BROWSE\_CO\_OP

Esta opção é válida apenas para filas locais, de alias e de modelo; não é válida para filas remotas, listas de distribuições e objetos que não são filas.

## **MQOO\_INQUIRE**

Abrir objeto para consultar atributos.

A fila, a lista de nomes, a definição de processo ou o gerenciador de filas é aberto para uso com chamadas MQINQ subsequentes.

Esta opção é válida para todos os tipos de objetos que não sejam listas de distribuição. Não será válido se `ObjectQMgrName` for o nome de um alias do gerenciador de filas; isso será verdadeiro mesmo se o valor do atributo **RemoteQMgrName** na definição local de uma fila remota usada para alias do gerenciador de filas for o nome do gerenciador de filas locais.

## **MQOO\_SET**

Abra a fila para configurar atributos

A fila é aberta para uso com chamadas MQSET subsequentes.

Esta opção é válida para todos os tipos de filas que não sejam listas de distribuição. Não será válido se `ObjectQMgrName` for o nome de uma definição local de uma fila remota; isso é verdadeiro mesmo se o valor do atributo **RemoteQMgrName** na definição local de uma fila remota usada para alias do gerenciador de filas for o nome do gerenciador de filas locais.

**Opções de ligação:** As opções a seguir se aplicam quando o objeto que está sendo aberto é uma fila de clusters; essas opções controlam a ligação do identificador de filas para uma instância da fila de clusters:

## **MQOO\_BIND\_ON\_OPEN**

O gerenciador de filas locais liga o identificador de filas a uma instância da fila de destino quando a fila é aberta. Como resultado, todas as mensagens colocadas usando esse identificador são enviadas para a mesma instância da fila de destino e pela mesma rota.

Essa opção é válida somente para filas e afeta somente filas de clusters. Se especificada para uma fila que não seja uma fila de cluster, a opção será ignorada.

## **MQOO\_BIND\_NOT\_FIXED**

Isso para o gerenciador de filas locais que liga a manipulação de filas a uma instância da fila de destino. Como resultado, chamadas MQPUT sucessivas usando essa manipulação enviam as mensagens para diferentes instâncias da fila de destino ou para a mesma instância, mas por rotas diferentes.. Ele também permite que a instância selecionada seja alterada posteriormente pelo gerenciador de filas locais, por um gerenciador de filas remotas ou por um agente de canal de mensagens (MCA), de acordo com as condições de rede...

**Nota:** Os aplicativos cliente e servidor que precisam trocar uma série de mensagens para concluir uma transação não devem usar `MQOO_BIND_NOT_FIXED` (ou `MQOO_BIND_AS_Q_DEF` quando o `DefBind` tiver o valor `MQBND_BIND_NOT_FIXED`), porque mensagens sucessivas na série podem ser enviadas para diferentes instâncias do aplicativo do servidor.

Se `MQOO_BROWSE` ou uma das opções `MQOO_INPUT_*` for especificada para uma fila de clusters, o gerenciador de filas será forçado a selecionar a instância local da fila de clusters.. Como resultado, a ligação do identificador de filas é corrigida, mesmo se `MQOO_BIND_NOT_FIXED` for especificado.

Se `MQOO_INQUIRE` for especificado com `MQOO_BIND_NOT_FIXED`, chamadas MQINQ sucessivas usando essa manipulação poderão consultar diferentes instâncias da fila de clusters, embora geralmente todas as instâncias tenham os mesmos valores de atributo..

`MQOO_BIND_NOT_FIXED` é válido apenas para filas e afeta apenas filas de clusters. Se especificada para uma fila que não seja uma fila de cluster, a opção será ignorada.

## **MQOO\_BIND\_ON\_GROUP**

Permite que um aplicativo solicite que um grupo de mensagens seja alocado na mesma instância de destino.

Essa opção é válida somente para filas e afeta somente filas de clusters. Se especificada para uma fila que não seja uma fila de cluster, a opção será ignorada.

## **MQOO\_BIND\_AS\_Q\_DEF**

O gerenciador de filas locais liga o identificador de filas na maneira definida pelo atributo da fila **DefBind**. O valor desse atributo é MQBND\_BIND\_ON\_OPEN, MQPNB\_BIND\_NOT\_FIXED ou MQBND\_BIND\_ON\_GROUP.

MQOO\_BIND\_AS\_Q\_DEF é o padrão quando MQOO\_BIND\_ON\_OPEN, MQOO\_BIND\_NOT\_FIXED, ou MQOO\_BIND\_ON\_GROUP não é especificado.

MQOO\_BIND\_AS\_Q\_DEF ajuda a documentação do programa. Não se pretende que essa opção seja usada com qualquer uma das outras duas opções de ligação, mas, como seu valor é zero, esse uso não pode ser detectado.

**Opções de contexto:** As opções a seguir controlam o processamento do contexto da mensagem:

## **MQOO\_SAVE\_ALL\_CONTEXT**

As informações de contexto estão associadas a este identificador de filas. Essas informações são configuradas a partir do contexto de qualquer mensagem recuperada utilizando esse identificador. Para obter mais informações sobre o contexto de mensagem, consulte [Contexto da mensagem](#) e [Informações de contexto de controle](#).

Essas informações de contexto podem ser transmitidas para uma mensagem que é, então, colocada em uma fila usando as chamadas MQPUT ou MQPUT1. Consulte as opções MQPMO\_PASS\_IDENTITY\_CONTEXT e MQPMO\_PASS\_ALL\_CONTEXT descritas em [“MQPMO-Opções de mensagem de colocação”](#) na página 518..

Até que uma mensagem tenha sido recuperada com êxito, o contexto não pode ser transmitido para uma mensagem que está sendo colocada em uma fila.

Uma mensagem recuperada usando uma das opções de navegação MQGMO\_BROWSE\_\* não tem suas informações de contexto salvas (embora os campos de contexto no parâmetro **MsgDesc** sejam configurados após uma procura)..

Esta opção é válida apenas para filas locais, de alias e de modelo; não é válida para filas remotas, listas de distribuições e objetos que não são filas. Uma das opções MQOO\_INPUT\_\* deve ser especificada.

## **MQOO\_PASS\_IDENTITY\_CONTEXT**

Isso permite que a opção MQPMO\_PASS\_IDENTITY\_CONTEXT seja especificada no parâmetro **PutMsgOpts** quando uma mensagem é colocada na fila; isso fornece à mensagem as informações do contexto de identidade de uma fila de entrada que foi aberta com a opção MQOO\_SAVE\_ALL\_CONTEXT. Para obter mais informações sobre o contexto de mensagem, consulte [Contexto da mensagem](#) e [Informações de contexto de controle](#).

A opção MQOO\_OUTPUT deve ser especificada.

Esta opção é válida para todos os tipos da fila, incluindo as listas de distribuição.

## **MQOO\_PASS\_ALL\_CONTEXT**

Isso permite que a opção MQPMO\_PASS\_ALL\_CONTEXT seja especificada no parâmetro **PutMsgOpts** quando uma mensagem é colocada na fila; isso fornece à mensagem as informações de contexto de identidade e origem de uma fila de entrada que foi aberta com a opção MQOO\_SAVE\_ALL\_CONTEXT. Para obter mais informações sobre o contexto de mensagem, consulte [Contexto da mensagem](#) e [Informações de contexto de controle](#).

Essa opção implica em MQOO\_PASS\_IDENTITY\_CONTEXT, que não precisa, portanto, ser especificado. A opção MQOO\_OUTPUT deve ser especificada.

Esta opção é válida para todos os tipos da fila, incluindo as listas de distribuição.

## **MQOO\_SET\_IDENTITY\_CONTEXT**

Isso permite que a opção MQPMO\_SET\_IDENTITY\_CONTEXT seja especificada no parâmetro **PutMsgOpts** quando uma mensagem é colocada em uma fila; isso fornece à mensagem as informações do contexto de identidade contidas no parâmetro **MsgDesc** especificado na chamada MQPUT ou MQPUT1. Para obter mais informações sobre o contexto de mensagem, consulte [Contexto da mensagem](#) e [Informações de contexto de controle](#).



Essa opção implica em `MQOO_PASS_IDENTITY_CONTEXT`, que não precisa, portanto, ser especificado. A opção `MQOO_OUTPUT` deve ser especificada.

Esta opção é válida para todos os tipos da fila, incluindo as listas de distribuição.

### **MQOO\_SET\_ALL\_CONTEXT**

Isso permite que a opção `MQPMO_SET_ALL_CONTEXT` seja especificada no parâmetro **PutMsgOpts** quando uma mensagem é colocada em uma fila; isso fornece à mensagem as informações de contexto de identidade e origem contidas no parâmetro **MsgDesc** especificado na chamada `MQPUT` ou `MQPUT1`. Para obter mais informações sobre o contexto de mensagem, consulte [Contexto da mensagem](#) e [Informações de contexto de controle](#).

Essa opção implica as seguintes opções, que não precisam, portanto, ser especificadas:

- `MQOO_PASS_IDENTITY_CONTEXT`
- `MQOO_PASS_ALL_CONTEXT`
- `MQOO_SET_IDENTITY_CONTEXT`

A opção `MQOO_OUTPUT` deve ser especificada.

Esta opção é válida para todos os tipos da fila, incluindo as listas de distribuição.

### **Opções de leitura antecipada:**

Ao chamar `MQOPEN` com `MQOO_READ_AHEAD`, o cliente IBM MQ somente ativará o modo leia mais adiante se determinadas condições forem atendidas. Essas condições incluem:

- O aplicativo cliente deve ser compilado e vinculado em relação as bibliotecas encadeadas do cliente IBM MQ MQI.
- O canal do cliente deve estar utilizando o protocolo TCP/IP.
- O canal deve ter a configuração de `SharingConversations` diferente de zero (`SHARECNV`) em ambas as definições de canal, do cliente e do servidor.

As opções a seguir controlam se mensagens não persistentes são enviadas ao cliente antes que um aplicativo as solicite. As notas a seguir se aplicam às opções de leitura antecipada:

- Apenas uma dessas opções pode ser especificada..
- Essas opções são válidas apenas para filas locais, de alias e de modelo. Eles não são válidos para filas remotas, listas de distribuição, tópicos ou gerenciadores de fila..
- Essas opções são aplicáveis apenas quando uma das opções `MQOO_BROWSE`, `MQOO_INPUT_SHARED` e `MQOO_INPUT_EXCLUSIVE` também são especificadas, embora não seja um erro especificar essas opções com `MQOO_INQUIRE` ou `MQOO_SET`.
- Se o aplicativo não estiver executando como um cliente IBM MQ, essas opções serão ignoradas.

### **MQOO\_NO\_READ\_AHEAD**

Mensagens não persistentes não são enviadas ao cliente antes que um aplicativo as solicite.

### **MQOO\_READ\_AHEAD**

Mensagens não persistentes são enviadas ao cliente antes que um aplicativo as solicite.

### **MQOO\_READ\_AHEAD\_AS\_Q\_DEF**

O comportamento de leitura antecipada é determinado pelo atributo de leitura antecipada padrão da fila sendo aberta. Esse é o valor-padrão.

**Outras opções:** As opções a seguir controlam a verificação de autorização, o que acontece quando o gerenciador de filas está em quiesce, se deve resolver o nome da fila local e multicast:

### **MQOO\_ALTERNATE\_USER\_AUTHORITY**

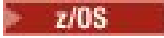
O campo `AlternateUserId` no parâmetro **ObjDesc** contém um identificador de usuário para usar para validar essa chamada `MQOPEN`. A chamada poderá ser bem-sucedida apenas se esse `AlternateUserId` estiver autorizado a abrir o objeto com as opções de acesso especificadas, independentemente de o identificador de usuário sob o qual o aplicativo está em execução estar autorizado a fazer isso. No entanto, isso não se aplica a nenhuma das opções de contexto.

especificadas, que são sempre verificadas com relação ao identificador de usuário sob o qual o aplicativo está em execução..

Esta opção é válida para todos os tipos de objeto

### **MQOO\_FAIL\_IF QUIESCING**

A chamada MQOPEN falhará se o gerenciador de filas estiver no estado de quiesce

 No z/OS, para um aplicativo CICS ou IMS , essa opção também força a falha da chamada MQOPEN se a conexão estiver no estado de quiesce

Esta opção é válida para todos os tipos de objeto

Para obter informações sobre canais clientes, consulte [IBM MQ MQI clients](#).

### **MQOO\_RESOLVE\_LOCAL\_Q**

Preencha ResolvedQName na estrutura MQOD com o nome da fila local que foi aberta. Da mesma forma, o Nome ResolvedQMgré preenchido com o nome do gerenciador de filas locais que hospeda a fila local.. Se a estrutura MQOD for menor que a Versão 3, MQOO\_RESOLVE\_LOCAL\_Q será ignorado sem nenhum erro retornado.

A fila local é sempre retornada quando uma fila local, de alias ou de modelo é aberta, mas este não é o caso quando, por exemplo, uma fila remota ou uma fila de cluster não local é aberta sem a opção MQOO\_RESOLVE\_LOCAL\_Q; o ResolvedQName e o ResolvedQMgré são preenchidos com o RemoteQName e o RemoteQMgré localizado na definição de fila remota ou de forma semelhante com a fila de cluster remoto escolhida.

Se você especificar MQOO\_RESOLVE\_LOCAL\_Q ao abrir uma fila remota, por exemplo, ResolvedQName é a fila de transmissão na qual as mensagens são colocadas. O Nome ResolvedQMgré preenchido com o nome do gerenciador de filas locais que hospeda a fila de transmissão.

Se você estiver autorizado para procura, entrada ou saída em uma fila, terá a autoridade necessária para especificar essa sinalização na chamada MQOPEN. Nenhuma autoridade especial é necessária.

Essa opção é válida apenas para filas e gerenciadores de fila

### **MQOO\_RESOLVE\_LOCAL\_TOPIC**

Preencha ResolvedQName na estrutura MQOD com o nome do tópico administrativo aberto.

### **MQOO\_NO\_MULTICAST**

As mensagens de publicação não são enviadas usando multicast

Essa opção é válida apenas com a opção MQOO\_OUTPUT Se for especificado sem MQOO\_OUTPUT, MQOPEN retorna com MQRC\_OPTIONS\_ERROR.

Esta opção é válida apenas para um tópico

### **Hobj**

Tipo: MQHOBJ - saída

Essa manipulação representa o acesso que foi estabelecido para o objeto Ele deve ser especificado em chamadas subsequentes do IBM MQ que operam no objeto Ele deixa de ser válido quando a chamada MQCLOSE é emitida ou quando a unidade de processamento que define o escopo da manipulação é finalizada

O escopo do identificador de objeto retornado é igual ao escopo do identificador de conexão especificado na chamada. Consulte [MQCONN-Hconn parameter](#) para obter informações sobre o escopo de manipulação.

### **CompCode**

Tipo: MQLONG - saída

O código de conclusão; é um dos seguintes:

**MQCC\_OK**

Indica conclusão bem-sucedida.

**MQCC\_WARNING**

Aviso (conclusão parcial).

**MQCC\_FAILED**

A chamada falhou.

**Razão**

Tipo: MQLONG - saída

O código de razão que qualifica *CompCode*.

Se *CompCode* for MQCC\_OK:

**MQRC\_NONE**

(0, X'000') Nenhuma razão a relatar.

Se *CompCode* for MQCC\_WARNING:

**MQRC\_MULTIPLE\_REASONS**

(2136, X'858 ') Códigos de razão múltiplos retornados.

Se *CompCode* for MQCC\_FAILED:

**MQRC\_ADAPTER\_NOT\_AVAILABLE**

(2204, X'89C') Adaptador não disponível.

**MQRC\_ADAPTER\_SERV\_LOAD\_ERROR**

(2130, X'852') Não foi possível carregar o módulo de serviço do adaptador.

**MQRC\_ALIAS\_BASE\_Q\_TYPE\_ERROR**

(2001, X'7D1') A fila base do alias não é um tipo válido.

**MQRC\_API\_EXIT\_ERROR**

(2374, X'946') Falha na saída de API.

**MQRC\_API\_EXIT\_LOAD\_ERROR**

(2183, X'887') Não foi possível carregar a saída de API.

**MQRC\_ASID\_MISMATCH**

(2157, X'86D') ASIDs de início e primárias são diferentes.

**MQRC\_CALL\_IN\_PROGRESS**

(2219, X'8AB') Chamada MQI inserida antes da chamada anterior concluída.

**MQRC\_CF\_NOT\_AVAILABLE**

(2345, X' 929 ') Recurso de acoplamento não disponível.

**MQRC\_CF\_STRUC\_AUTH\_FAILED**

(2348, X'92C') A verificação de autorização da estrutura do recurso de acoplamento falhou..

**MQRC\_CF\_STRUC\_ERROR**

(2349, X'92D') Estrutura de recurso de acoplamento não válida.

**MQRC\_CF\_STRUC\_FAILED**

(2373, X'945') Falha na estrutura do recurso de acoplamento.

**MQRC\_CF\_STRUC\_IN\_USE**

(2346, X'92A') Estrutura de recurso de acoplamento em uso.

**MQRC\_CF\_STRUC\_LIST\_HDR\_IN\_USE**

(2347, X'92B') Cabeçalho da lista da estrutura do recurso de acoplamento em uso.

**MQRC\_CICS\_WAIT\_FAILED**

(2140, X'85C') Pedido de espera rejeitado por CICS.

**MQRC\_CLUSTER\_EXIT\_ERROR**

(2266, X'8DA') Falha na saída de carga de trabalho do cluster.

**MQRC\_CLUSTER\_PUT\_INIBIDA**

(2268, X'8DC') Chamadas Put inibidas para todas as filas no cluster.

**MQRC\_CLUSTER\_RESOLUTION\_ERROR**  
(2189, X'88D') Falha na resolução do nome do cluster.

**MQRC\_CLUSTER\_RESOURCE\_ERROR**  
(2269, X'8DD') Erro de recurso de cluster

**MQRC\_CONNECTION\_BROKEN**  
(2009, X'7D9') Conexão ao gerenciador de filas perdida.

**MQRC\_CONNECTION\_NOT\_AUTHORIZED**  
(2217, X'8A9') Não autorizado para conexão.

**MQRC\_CONNECTION QUIESCING**  
(2202, X'89A') Quiesce de conexão.

**MQRC\_CONNECTION\_STOPPING**  
(2203, X'89B') Conexão sendo encerrada.

**MQRC\_DB2\_NOT\_AVAILABLE**  
(2342, X' 926 ') Db2 subsistema não disponível.

**MQRC\_DEF\_XMIT\_Q\_TYPE\_ERROR**  
(2198, X'896 ') Fila de transmissão padrão não local.

**MQRC\_DEF\_XMIT\_Q\_USAGE\_ERROR**  
(2199, X'897 ') Erro de uso da fila de transmissão padrão

**MQRC\_DYNAMIC\_Q\_NAME\_ERROR**  
(2011, X'7DB') Nome da fila dinâmica inválido.

**MQRC\_HANDLE\_NOT\_AVAILABLE**  
(2017, X'7E1') Não há mais identificadores disponíveis

**MQRC\_HCONN\_ERROR**  
(2018, X'7E2') Manipulação de conexões não válida.

**MQRC\_HOBJ\_ERROR**  
(2019, X'7E3') Manipulação de objetos não válida.

**MQRC\_MULTIPLE\_REASONS**  
(2136, X'858 ') Códigos de razão múltiplos retornados.

**MQRC\_NAME\_IN\_USE**  
(2201, X'899 ') Nome em uso.

**MQRC\_NAME\_NOT\_VALID\_FOR\_TYPE**  
(2194, X'892 ') Nome do objeto inválido para o tipo de objeto.

**MQRC\_NOT\_AUTHORIZED**  
(2035, X'7F3') Não autorizado para acesso.

**MQRC\_OBJECT\_ALREADY\_EXISTS**  
(2100, X'834 ') O objeto existe.

**MQRC\_OBJECT\_DAMAGED**  
(2101, X'835') Objeto danificado.

**MQRC\_OBJECT\_IN\_USE**  
(2042, X'7FA') Objeto já aberto com opções conflitantes.

**MQRC\_OBJECT\_LEVEL\_INCOMPATÍVEL**  
(2360, X' 938 ') Nível de objeto não compatível.

**MQRC\_OBJECT\_NAME\_ERROR**  
(2152, X'868 ') Nome do objeto inválido.

**MQRC\_OBJECT\_NOT\_UNIQUE**  
(2343, X' 927 ') Objeto não exclusivo.

**MQRC\_OBJECT\_Q\_MGR\_NAME\_ERROR**  
(2153, X'869 ') O nome do gerenciador de filas de objeto não é válido.

**MQRC\_OBJECT\_RECORDS\_ERROR**  
(2155, X'86B') Registros de objeto inválidos.

**MQRC\_OBJECT\_STRING\_ERROR**

(2441, X'0989 ') O campo Objectstring não é válido

**MQRC\_OBJECT\_TYPE\_ERROR**

(2043, X'7FB') Tipo de objeto inválido.

**MQRC\_OD\_ERROR**

(2044, X'7FC') Estrutura do descritor de objeto inválida.

**MQRC\_OPTION\_NOT\_VALID\_FOR\_TYPE**

(2045, X'7FD') Opção não válida para tipo de objeto.

**MQRC\_OPTIONS\_ERROR**

(2046, X'7FE') Opções não válidas ou não consistentes.

**MQRC\_PAGESET\_ERROR**

(2193, X'891') Erro ao acessar o conjunto de dados do conjunto de páginas.

**MQRC\_PAGESET\_FULL**

(2192, X'890') O meio de armazenamento externo está cheio.

**MQRC\_Q\_DELETED**

(2052, X'804') A fila foi excluída.

**MQRC\_Q\_MGR\_NAME\_ERROR**

(2058, X'80A') Nome do gerenciador de filas não válido ou não conhecido.

**MQRC\_Q\_MGR\_NOT\_AVAILABLE**

(2059, X'80B') Gerenciador de filas não disponível para conexão.

**MQRC\_Q\_MGR QUIESCING**

(2161, X'871') Quiesce do gerenciador de filas.

**MQRC\_Q\_MGR\_STOPPING**

(2162, X'872') Gerenciador de filas sendo encerrado.

**MQRC\_Q\_TYPE\_ERROR**

(2057, X'809 ') Tipo de fila inválido.

**MQRC\_RECS\_PRESENT\_ERROR**

(2154, X'86A') Número de registros presentes não válidos.

**MQRC\_REMOTE\_Q\_NAME\_ERROR**

(2184, X'888 ') Nome da fila remota inválido.

**MQRC\_RESOURCE\_PROBLEM**

(2102, X'836') Recursos insuficientes do sistema disponíveis.

**MQRC\_RESPONSE\_RECORDS\_ERROR**

(2156, X'86C') Registros de resposta inválidos.

**MQRC\_SECURITY\_ERROR**

(2063, X'80F') Ocorreu um erro na segurança.

**MQRC\_SELECTOR\_SYNTAX\_ERROR**

2459 (X'099B') Uma chamada MQOPEN, MQPUT1 ou MQSUB foi emitida, mas foi especificada uma sequência de seleção que continha um erro de sintaxe.

**MQRC\_STOPPED\_BY\_CLUSTER\_EXIT**

(2188, X'88C') Chamada rejeitada pela saída de carga de trabalho do cluster.

**MQRC\_STORAGE\_MEDIUM\_FULL**

(2192, X'890') O meio de armazenamento externo está cheio.

**MQRC\_STORAGE\_NOT\_AVAILABLE**

(2071, X'817') Armazenamento insuficiente disponível.

**MQRC\_SUPPRESSED\_BY\_EXIT**

(2109, X'83D') Chamada suprimida pelo programa de saída.

**MQRC\_UNEXPECTED\_ERROR**

(2195, X'893') Ocorreu um erro inesperado.

**MQRC\_UNKNOWN\_ALIAS\_BASE\_Q**

(2082, X'822 ') Fila base de alias desconhecida.

**MQRC\_UNKNOWN\_DEF\_XMIT\_Q**

(2197, X'895 ') Fila de transmissão padrão desconhecida.

**MQRC\_UNKNOWN\_OBJECT\_NAME**

(2085, X'825 ') Nome do objeto desconhecido.

**MQRC\_UNKNOWN\_OBJECT\_Q\_MGR**

(2086, X'826 ') Gerenciador de filas de objeto desconhecido.

**MQRC\_UNKNOWN\_REMOTE\_Q\_MGR**

(2087, X'827 ') Gerenciador de filas remotas desconhecido.

**MQRC\_UNKNOWN\_XMIT\_Q**

(2196, X'894 ') Fila de transmissão desconhecida.

**MQRC\_ERRR\_CF\_LEVEL**

(2366, X'93E') A estrutura do recurso de acoplamento está no nível errado.

**MQRC\_XMIT\_Q\_TYPE\_ERROR**

(2091, X'82B') Fila de transmissão não local.

**MQRC\_XMIT\_Q\_USAGE\_ERROR**

(2092, X'82C') Fila de transmissão com uso errado.

Para obter informações detalhadas sobre esses códigos, consulte [Mensagens e códigos de razão](#).

## Notas gerais de utilização

1. O objeto aberto é um dos seguintes:

- Uma fila para:
  - Obter ou procurar mensagens (usando a chamada MQGET)
  - Mensagens put (usando a chamada MQPUT)
  - Consultar os atributos da fila (usando a chamada MQINQ)
  - Configure os atributos da fila (usando a chamada MQSET)..

Se a fila denominada for uma fila modelo, uma fila local dinâmica será criada. Consulte o parâmetro **ObjDesc** descrito em [“MQOPEN-Abrir objeto”](#) na página 759..

Uma lista de distribuição é um tipo especial de objeto de fila que contém uma lista de filas.. Pode ser aberto para colocar mensagens, mas não para obter ou procurar mensagens ou para consultar ou configurar atributos. Consulte a nota de uso 8 para obter detalhes adicionais

Uma fila que tem QSGDISP (GROUP) é um tipo especial de definição de fila que não pode ser usado com as chamadas MQOPEN ou MQPUT1

- Uma lista de nomes para consultar os nomes das filas na lista (usando a chamada MQINQ).
  - Uma definição de processo para consultar sobre os atributos de processo (usando a chamada MQINQ)
  - O gerenciador de filas para consultar sobre os atributos do gerenciador de fila local (usando a chamada MQINQ)
  - Um tópico para publicar uma mensagem (usando a chamada MQPUT)
2. Um aplicativo pode abrir o mesmo objeto mais de uma vez.. Uma manipulação de objetos diferente é retornada para cada abertura Cada identificador que é retornado pode ser usado para as funções para as quais a abertura correspondente foi executada.
3. Se o objeto que está sendo aberto for uma fila diferente de uma fila de clusters, toda a resolução do nome dentro do gerenciador de filas locais ocorrerá no momento da chamada MQOPEN Isso pode incluir:

- Resolução do nome de uma definição local de uma fila remota para o nome do gerenciador de fila remota e o nome pelo qual a fila é conhecida no gerenciador de filas remotas
- Resolução do nome do gerenciador de filas remotas para o nome de uma fila de transmissão local
- **z/OS** Somente no z/OS, a resolução do nome do gerenciador de filas remotas para o nome da fila de transmissão compartilhada usada por agente IGQ (será aplicada apenas se os gerenciadores de filas locais e remotos pertencerem ao mesmo grupo de filas compartilhadas)
- Resolução de alias para o nome de uma fila base ou objeto de tópico.

No entanto, esteja ciente de que as chamadas MQINQ ou MQSET subsequentes para o identificador estão relacionadas exclusivamente ao nome que foi aberto e não ao objeto resultante após a resolução do nome ter ocorrido. Por exemplo, se o objeto aberto for um alias, os atributos retornados pela chamada MQINQ serão os atributos do alias, não os atributos da fila de base ou um objeto de tópico para o qual o alias é resolvido

Se o objeto que está sendo aberto for uma fila de clusters, a resolução do nome poderá ocorrer no momento da chamada MQOPEN ou ser adiada até mais tarde... O ponto no qual a resolução ocorre é controlado pelas opções MQOO\_BIND\_\* especificadas na chamada MQOPEN:

- MQOO\_BIND\_ON\_OPEN
- MQOO\_BIND\_NOT\_FIXED
- MQOO\_BIND\_AS\_Q\_DEF
- MQOO\_BIND\_ON\_GROUP

Consulte [Resolução de nome](#) para obter mais informações sobre a resolução de nome para filas de cluster...

4. Uma chamada MQOPEN com a opção MQOO\_BROWSE estabelece um cursor de procura, para uso com chamadas MQGET que especificam a manipulação de objetos e uma das opções de procura. Isso permite que a fila seja digitalizada sem alterar seu conteúdo. Uma mensagem que foi localizada pela procura pode ser removida da fila usando a opção MQGMO\_MSG\_UNDER\_CURSOR.

Vários cursores de navegação podem estar ativos para um único aplicativo emitindo várias solicitações MQOPEN para a mesma fila..

5. Os aplicativos iniciados por um monitor acionador recebem o nome da fila associada ao aplicativo quando o aplicativo é iniciado. Esse nome da fila pode ser especificado no parâmetro **ObjDesc** para abrir a fila. Consulte [“MQTMC2 -Mensagem do acionador 2 \(formato de caractere\)”](#) na página 629 para obter detalhes adicionais.

## Opções de leitura antecipada

Ao chamar MQOPEN com MQOO\_READ\_AHEAD, o cliente IBM MQ somente ativará o modo leia mais adiante se determinadas condições forem atendidas. Essas condições incluem:

- O aplicativo cliente deve ser compilado e vinculado em relação as bibliotecas encadeadas do cliente IBM MQ MQI.
- O canal do cliente deve estar utilizando o protocolo TCP/IP
- O canal deve ter a configuração de SharingConversations diferente de zero (SHARECNV) em ambas as definições de canal, do cliente e do servidor.

As notas a seguir aplicam-se ao uso de opções de leitura antecipada.

1. As opções de leitura antecipada são aplicáveis apenas quando uma e apenas uma das opções MQOO\_BROWSE, MQOO\_INPUT\_SHARED e MQOO\_INPUT\_EXCLUSIVE também são especificadas. Um erro não será lançado se as opções de leitura antecipada forem especificadas com as opções MQOO\_INQUIRE ou MQOO\_SET
2. A leitura antecipada não será ativada quando solicitado se as opções usadas na primeira chamada MQGET não forem suportadas para uso com a leitura antecipada. Além disso, a leitura antecipada é

desativada quando o cliente está se conectando a um gerenciador de filas que não suporta leitura antecipada..

3. Se o aplicativo não estiver executando como um cliente IBM MQ , as opções de leitura antecipada serão ignoradas.

## Filas de Clusters

As seguintes notas se aplicam ao uso de filas de clusters.

1. Quando uma fila de clusters é aberta pela primeira vez e o gerenciador de filas locais não é um gerenciador de filas de repositório completo, o gerenciador de filas locais obtém informações sobre a fila de clusters de um gerenciador de filas de repositório completo. Quando a rede está ocupada, pode levar vários segundos para que o gerenciador de fila local receba as informações necessárias do gerenciador de filas do repositório. Como resultado, o aplicativo que emite a chamada MQOPEN pode ter que aguardar até 10 segundos antes que o controle retorne da chamada MQOPEN. Se o gerenciador de filas locais não receber as informações necessárias sobre a fila de clusters dentro desse tempo, a chamada falhará com o código de razão MQRC\_CLUSTER\_RESOLUTION\_ERROR

2. Quando uma fila de clusters é aberta e há várias instâncias da fila no cluster, a instância aberta depende das opções especificadas na chamada MQOPEN:

- Se as opções especificadas incluírem uma das seguintes:

- MQOO\_BROWSE
- MQOO\_INPUT\_AS\_Q\_DEF
- MQOO\_INPUT\_EXCLUSIVE
- MQOO\_INPUT\_SHARED
- MQOO\_SET

A instância da fila de clusters aberta deve ser a instância local. Se não houver instância local da fila, a chamada MQOPEN falhará.

- Se as opções especificadas não incluírem nenhuma das opções descritas anteriormente, mas incluírem uma ou ambas as seguintes:

- MQOO\_INQUIRE
- MQOO\_OUTPUT

a instância aberta é a instância local se houver uma e uma instância remota, caso contrário (se estiver usando os padrões CLWLUSEQ). A instância escolhida pelo gerenciador de fila pode, no entanto, ser alterada por uma saída de carga de trabalho do cluster (se houver uma)...

3. Se houver uma assinatura para a fila, mas não for reconhecida por um repositório completo, o objeto não estará presente no cluster e a chamada falhará com o código de razão MQRC\_OBJECT\_NAME.

Para obter mais informações sobre filas de clusters, consulte [Filas de clusters](#)..

## Listas de distribuição

As seguintes notas aplicam-se ao uso de listas de distribuição.

As listas de distribuição são suportadas nos seguintes ambientes:

-  AIX
-  IBM i
-  Linux
-  Windows

e para o IBM MQ MQI clients conectado a esses sistemas



1. Os campos na estrutura MQOD devem ser configurados conforme a seguir ao abrir uma lista de distribuição:
  - Version deve ser MQOD\_VERSION\_2 ou maior.
  - ObjectType deve ser MQOT\_Q..
  - ObjectName deve estar em branco ou a sequência nula.
  - ObjectQMgrName deve estar em branco ou a sequência nula.
  - RecsPresent Deve ser maior que zero.
  - Um de ObjectRecOffset e ObjectRecPtr deve ser zero e o outro diferente de zero.
  - No máximo um de ResponseRecOffset e ResponseRecPtr pode ser diferente de zero.
  - Deve haver registros de objeto RecsPresent , endereçados por ObjectRecOffset ou ObjectRecPtr Os registros do objeto devem ser configurados para os nomes das filas de destino a serem abertas
  - Se um de ResponseRecOffset e ResponseRecPtr for diferente de zero, deverá haver RecsPresent registros de resposta presentes... Eles serão configurados pelo gerenciador de filas se a chamada for concluída com o código de razão MQRC\_MULTIPLE\_REASON

Um MQOD version-2 também pode ser usado para abrir uma única fila que não esteja em uma lista de distribuição, assegurando que RecsPresent seja zero.

2. Apenas as opções de abertura a seguir são válidas no parâmetro **Options** :
  - MQOO\_OUTPUT
  - MQOO\_PASS\_ \* \_CONTEXT
  - MQOO\_SET\_ \* \_CONTEXT
  - MQOO\_ALTERNATE\_USER\_AUTHORITY
  - MQOO\_FAIL\_IF\_QUIESCING
3. As filas de destino na lista de distribuição podem ser filas locais, de alias ou remotas, mas não podem ser filas modelo. Se uma fila modelo for especificada, essa fila falhará ao abrir, com código de razão MQRC\_Q\_TYPE\_ERROR. No entanto, isso não evita que outras filas na lista sejam abertas com êxito
4. Os parâmetros de código de conclusão e de código de razão são configurados conforme a seguir::
  - Se as operações de abertura para as filas na lista de distribuição todas forem bem-sucedidas ou falharem da mesma maneira, os parâmetros de código de conclusão e de código de razão serão configurados para descrever o resultado comum Os registros de resposta MQRR (se fornecidos pelo aplicativo) não são configurados neste caso.
 

Por exemplo, se cada abertura for bem-sucedida, o código de conclusão será configurado como MQCC\_OK e o código de razão será configurado como MQRC\_NONE; se cada abertura falhar porque nenhuma das filas existe, os parâmetros serão configurados como MQCC\_FAILED e MQRC\_UNKNOWN\_OBJECT\_NAME.
  - Se as operações de abertura para as filas na lista de distribuição não forem todas bem-sucedidas ou falharem da mesma maneira:
    - O parâmetro do código de conclusão será configurado como MQCC\_WARNING se pelo menos uma abertura for bem-sucedida e como MQCC\_FAILED se todos falharem.
    - O parâmetro do código de razão é configurado como MQRC\_MULTIPLE\_REASON
    - Os registros de resposta (se fornecidos pelo aplicativo) são configurados para os códigos de conclusão individuais e os códigos de razão para as filas na lista de distribuição
5. Quando uma lista de distribuição tiver sido aberta com êxito, o identificador Hobj retornado pela chamada poderá ser usado em chamadas MQPUT subsequentes para colocar mensagens em filas na lista de distribuição e em uma chamada MQCLOSE para abrir mão do acesso à lista de distribuição A única opção de fechamento válida para uma lista de distribuições é MQCO\_NONE

A chamada MQPUT1 também pode ser usada para colocar uma mensagem em uma lista de distribuição; a estrutura MQOD que define as filas na lista é especificada como um parâmetro nessa chamada..

6. Cada destino aberto com sucesso na lista de distribuição conta como um identificador separado ao verificar se o aplicativo excedeu o número máximo permitido de identificadores (consulte o atributo do gerenciador de filas **MaxHandles** ). Isso é verdadeiro mesmo quando dois ou mais destinos na lista de distribuição são resolvidos para a mesma fila física. Se a chamada MQOPEN ou MQPUT1 para uma lista de distribuição causar o número de identificadores em uso pelo aplicativo para exceder MaxHandles, a chamada falhará com o código de razão MQRC\_HANDLE\_NOT\_AVAILABLE.
7. Cada destino aberto com sucesso tem o valor de seu atributo **OpenOutputCount** incrementado em um. Se dois ou mais destinos na lista de distribuição forem resolvidos para a mesma fila física, essa fila terá seu atributo **OpenOutputCount** incrementado pelo número de destinos na lista de distribuição que são resolvidos para essa fila..
8. Qualquer mudança nas definições de fila que teria feito com que uma manipulação se tornasse inválida se as filas fossem abertas individualmente (por exemplo, uma mudança no caminho de resolução) não faz com que a manipulação da lista de distribuição se torne inválida. No entanto, isso resulta em uma falha para essa fila específica quando a manipulação da lista de distribuições é usada em uma chamada MQPUT subsequente
9. Uma lista de distribuição pode conter apenas um destino.

## Filas Remotas

As seguintes notas se aplicam ao uso de filas remotas.

Uma fila remota pode ser especificada de uma de duas maneiras no parâmetro **ObjDesc** desta chamada

- Especificando para **ObjectName** o nome de uma definição local da fila remota. Nesse caso, **ObjectQMgrName** refere-se ao gerenciador de filas locais e pode ser especificado como espaços em branco ou (na linguagem de programação C) uma sequência nula.

A validação de segurança executada pelo gerenciador de fila local verifica se o usuário está autorizado a abrir a definição local da fila remota

- Especificando para **ObjectName** o nome da fila remota, conforme conhecido para o gerenciador de filas remotas. Nesse caso, **ObjectQMgrName** é o nome do gerenciador de fila remoto

A validação de segurança executada pelo gerenciador de filas locais verifica se o usuário está autorizado a enviar mensagens para a fila de transmissão resultante do processo de resolução de nome

Em ambos os casos:

- Nenhuma mensagem é enviada pelo gerenciador de filas locais ao gerenciador de filas remotas para verificar se o usuário está autorizado a colocar mensagens na fila.
- Quando uma mensagem chega ao gerenciador de filas remotas, o gerenciador de filas remotas pode rejeitá-lo porque o usuário que está originando a mensagem não está autorizado.

Consulte os campos **ObjectName** e **ObjectQMgrName** descritos em [“MQOD-Descrição de objetos”](#) na página 497 , para obter mais informações

## Objetos

### Segurança


As notas a seguir estão relacionadas aos aspectos de segurança do uso do MQOPEN.

O gerenciador de filas executa verificações de segurança quando uma chamada MQOPEN é emitida, para verificar se o identificador de usuário sob o qual o aplicativo está em execução possui o nível apropriado de autoridade antes que o acesso seja permitido. A verificação de autoridade é feita no nome do objeto que está sendo aberto e não no nome, ou nomes, resultante após um nome ter sido resolvido.

Se o objeto que está sendo aberto for uma fila de alias que aponta para um objeto do tópico, o gerenciador de filas executará uma verificação de segurança no nome da fila de alias, antes de executar uma verificação de segurança para o tópico como se o objeto do tópico tivesse sido usado diretamente

Se o objeto que está sendo aberto for um objeto do tópico, seja com `ObjectName` sozinho ou usando o `ObjectString` (com ou sem uma base `ObjectName`), o gerenciador de filas executará a verificação de segurança usando a sequência de tópicos resultante, obtida de dentro do objeto do tópico especificado em `ObjectName`, se necessário, concatenando-a com aquela fornecida em `ObjectString`, em seguida, localizando o objeto do tópico mais próximo nesse ponto ou acima na árvore de tópicos para executar a verificação de segurança. Esse pode não ser o mesmo objeto do tópico especificado em `ObjectName`.


Se o objeto que está sendo aberto for uma fila modelo, o gerenciador de filas executará uma verificação de segurança completa com relação ao nome da fila modelo e ao nome da fila dinâmica criada. Se a fila dinâmica resultante for aberta explicitamente, uma verificação de segurança de recurso adicional será executada com relação ao nome da fila dinâmica.

 No z/OS, o gerenciador de filas executará verificações de segurança somente se a segurança estiver ativada Para obter mais informações sobre a verificação de segurança, consulte [Configurando a segurança em z/OS](#)

## Atributos

As notas a seguir estão relacionadas a atributos.

Os atributos de um objeto podem ser alterados enquanto um aplicativo tiver o objeto aberto Em muitos casos, o aplicativo não percebe isso, mas para determinados atributos, o gerenciador de filas marca o identificador como não mais válido Esses atributos são:

- Qualquer atributo que afete a resolução do nome do objeto Isso se aplica independentemente das opções de abertura usadas e inclui o seguinte:
  - Uma mudança para o atributo **BaseQName** de uma fila de alias que está aberta
  - Uma mudança para o atributo **TargetType** de uma fila de alias que está aberta
  - Uma mudança nos atributos de fila **RemoteQName** ou **RemoteQMgrName** , para qualquer identificador que esteja aberto para essa fila ou para uma fila que seja resolvida por meio dessa definição como um alias do gerenciador de filas
  - Qualquer mudança que faz com que uma manipulação atualmente aberta para uma fila remota seja resolvida para uma fila de transmissão diferente ou falhe em resolver para uma. Por exemplo, isso pode incluir:
    - Uma mudança no atributo **XmitQName** da definição local de uma fila remota, independentemente de a definição estar sendo usada para uma fila ou para um alias do gerenciador de filas
    -  Somente no z/OS , uma mudança para o valor do atributo do gerenciador de filas **IntraGroupqueuing** ou uma mudança na definição da fila de transmissão compartilhada (SYSTEM.QSG.TRANSMIT.QUEUE) usado pelo agente IGQ.

Há uma exceção para isso: a criação de uma nova fila de transmissão.. Um identificador que teria sido resolvido para esta fila se estivesse presente quando o identificador foi aberto, mas, em vez disso, resolvido para a fila de transmissão padrão, não é inválido.

- Uma mudança no atributo do gerenciador de filas do **DefXmitQName** Nesse caso, todos os identificadores abertos que foram resolvidos para a fila nomeada anteriormente (que foram resolvidos para ela apenas porque era a fila de transmissão padrão) são marcados como inválidos. As manipulações que foram resolvidas para essa fila por outras razões não são afetada
- O atributo da fila **Shareability** , se houver dois ou mais identificadores que estão atualmente fornecendo acesso `MQOO_INPUT_SHARED` para essa fila, ou para uma fila que é resolvida para essa fila, Nesse caso, todos os identificadores que estão abertos para essa fila ou para uma fila que é resolvida para essa fila serão marcados como inválidos, independentemente das opções de abertura.

**z/OS** No z/OS, os identificadores descritos anteriormente serão marcados como inválidos se um ou mais identificadores estiverem atualmente fornecendo acesso MQOO\_INPUT\_SHARED ou MQOO\_INPUT\_EXCLUSIVE à fila..

- O atributo da fila **Usage**, para todas as manipulações que estão abertas para esta fila ou para uma fila que é resolvida para esta fila, independentemente das opções de abertura

Quando um identificador é marcado como inválido, todas as chamadas subsequentes (diferentes de MQCLOSE) usando esse identificador falham com o código de razão MQRC\_OBJECT\_CHANGED. O aplicativo deve emitir uma chamada MQCLOSE (usando o identificador original), em seguida, reabrir a fila. Quaisquer atualizações não confirmadas com relação à manipulação antiga de chamadas bem-sucedidas anteriores ainda podem ser confirmadas ou restauradas, conforme necessário pela lógica de aplicativo. Se a mudança de um atributo causar isso, use uma versão de força especial da chamada.

## Chamada C

```
MQOPEN (Hconn, &ObjDesc, Options, &Hobj, &CompCode,
        &Reason);
```

Declare os parâmetros conforme a seguir:

```
MQHCONN  Hconn;      /* Connection handle */
MQOD     ObjDesc;    /* Object descriptor */
MQLONG   Options;    /* Options that control the action of MQOPEN */
MQHOBJ   Hobj;       /* Object handle */
MQLONG   CompCode;   /* Completion code */
MQLONG   Reason;     /* Reason code qualifying CompCode */
```

## Chamada COBOL

```
CALL 'MQOPEN' USING HCONN, OBJDESC, OPTIONS, HOBJ, COMPCODE, REASON
```

Declare os parâmetros conforme a seguir:

```
** Connection handle
01 HCONN      PIC S9(9) BINARY.
** Object descriptor
01 OBJDESC.
   COPY CMQODV.
** Options that control the action of MQOPEN
01 OPTIONS    PIC S9(9) BINARY.
** Object handle
01 HOBJ       PIC S9(9) BINARY.
** Completion code
01 COMPCODE   PIC S9(9) BINARY.
** Reason code qualifying COMPCODE
01 REASON     PIC S9(9) BINARY.
```

## Chamada PL/I

```
call MQOPEN (Hconn, ObjDesc, Options, Hobj, CompCode, Reason);
```

Declare os parâmetros conforme a seguir:

```
dcl Hconn      fixed bin(31); /* Connection handle */
dcl ObjDesc    like MQOD;    /* Object descriptor */
dcl Options    fixed bin(31); /* Options that control the action of
                               MQOPEN */
dcl Hobj       fixed bin(31); /* Object handle */
```

```
dcl CompCode fixed bin(31); /* Completion code */
dcl Reason fixed bin(31); /* Reason code qualifying CompCode */
```

## Chamada do Assembler de Alto Nível

```
CALL MQOPEN, (HCONN, OBJDESC, OPTIONS, HOBJ, COMPCODE, REASON)
```

Declare os parâmetros conforme a seguir:

HCONN	DS	F	Connection handle
OBJDESC	CMQODA	,	Object descriptor
OPTIONS	DS	F	Options that control the action of MQOPEN
HOBJ	DS	F	Object handle
COMPCODE	DS	F	Completion code
REASON	DS	F	Reason code qualifying COMPCODE

## Chamada do Visual Basic

Windows

```
MQOPEN Hconn, ObjDesc, Options, Hobj, CompCode, Reason
```

Declare os parâmetros conforme a seguir:

```
Dim Hconn As Long 'Connection handle'
Dim ObjDesc As MQOD 'Object descriptor'
Dim Options As Long 'Options that control the action of MQOPEN'
Dim Hobj As Long 'Object handle'
Dim CompCode As Long 'Completion code'
Dim Reason As Long 'Reason code qualifying CompCode'
```

## MQPUT-Mensagem de colocação

A chamada MQPUT coloca uma mensagem em uma fila ou lista de distribuição ou em um tópico. A fila, lista de distribuição ou tópico já deve estar aberto.

### Sintaxe


MQPUT (*Hconn, Hobj, MsgDesc, PutMsg, BufferLength, Buffer, CompCode, Razão*)

### Parâmetros

#### Hconn

Tipo: MQHCONN - entrada

Esta manipulação representa a conexão ao gerenciador de filas. O valor de Hconn foi retornado por uma chamada anterior MQCONN ou MQCONNX.

 Em z/OS para aplicativos CICS, a chamada MQCONN pode ser omitida e o seguinte valor especificado para Hconn:

#### MQHC\_DEF\_HCONN

Manipulação de conexões padrão.

#### Hobj

Tipo: MQHOBJ - entrada

Esse identificador representa a fila na qual a mensagem é incluída, ou o tópico no qual a mensagem é publicada. O valor de Hobj foi retornado por uma chamada MQOPEN anterior que especificou a opção MQOO\_OUTPUT.

## MsgDesc

Tipo: MQMD - entrada/saída

Esta estrutura descreve os atributos da mensagem que está sendo enviada e recebe informações sobre a mensagem após o pedido de colocação ser concluído. Consulte [“MQMD - descritor de mensagem”](#) na página 435 para obter detalhes.

Se o aplicativo fornecer um MQMD version-1, os dados da mensagem poderão ser prefixados com uma estrutura MQMDE para especificar valores para os campos existentes no MQMD version-2, mas não no version-1. O campo *Formato* no MQMD deve ser configurado como MQFMT\_MD\_EXTENSION para indicar que um MQMDE está presente.. Consulte [“MQMDE-Extensão do descritor de mensagens”](#) na página 488 para obter mais detalhes.

O aplicativo não precisa fornecer uma estrutura MQMD se uma manipulação de mensagem válida for fornecida nos campos *OriginalMsgHandle* ou *NewMsgHandle* da estrutura MQPMO. Se nada for fornecido em um desses campos, o descritor da mensagem será obtido do descritor associado aos identificadores de mensagens..

Se você usar ou planejar usar as saídas de API, recomendamos que você forneça explicitamente uma estrutura MQMD e não use os descritores de mensagens associados às manipulações de mensagem. Isso ocorre porque a saída de API associada à chamada MQPUT ou MQPUT1 não pode determinar quais valores MQMD são usados pelo gerenciador de filas para concluir a solicitação MQPUT ou MQPUT1.

## PutMsgOpts

Tipo: MQPMO-entrada / saída

Consulte [“MQPMO-Opções de mensagem de colocação”](#) na página 518 para obter detalhes.

## BufferLength

Tipo: MQLONG - entrada

O comprimento da mensagem em *BufferZero* é válido e indica que a mensagem não contém nenhum dado do aplicativo. O limite superior para *BufferLength* depende de vários fatores:

- Se o destino for uma fila local ou for resolvido para uma fila local, o limite superior dependerá se:
  - O gerenciador de filas locais suporta a segmentação
  - O aplicativo de envio especifica a sinalização que permite que o gerenciador de filas segmente a mensagem. Essa sinalização é MQMF\_SEGMENTATION\_ALLOWED e pode ser especificada em um MQMD version-2 ou em um MQMDE usado com um MQMD version-1.

Se ambas as condições forem satisfeitas, *BufferLength* não poderá exceder 999 999 999 menos o valor do campo *Offset* no MQMD. Portanto, a mensagem lógica mais longa que pode ser colocada é 999 999 999 bytes (quando *Offset* é zero). No entanto, as restrições de recursos impostas pelo sistema operacional ou ambiente no qual o aplicativo está em execução podem resultar em um limite inferior.

Se uma ou ambas as condições anteriores não forem satisfeitas, o *BufferLength* não poderá exceder o atributo **MaxMsgLength** da fila e o atributo **MaxMsgLength** do gerenciador de filas.

- Se o destino for uma fila remota ou for resolvido para uma fila remota, as condições para filas locais se aplicam, mas em cada gerenciador de filas por meio do qual a mensagem deve passar para atingir a fila de destino; em especial:
  1. A fila de transmissão local usada para armazenar a mensagem temporariamente no gerenciador de filas locais
  2. Filas de transmissão intermediárias (se houver) utilizadas para armazenar a mensagem nos gerenciadores de fila na rota entre os gerenciadores de filas local e de destino
  3. A fila de destino no gerenciador de filas de destino

A mensagem mais longa que pode ser colocada é, portanto, controlada pelo mais restritivo dessas filas e gerenciadores de filas.

Quando uma mensagem está em uma fila de transmissão, informações adicionais residem nos dados da mensagem e isso reduz a quantidade de dados do aplicativo que podem ser transportados. Nessa situação, subtraia os bytes MQ\_MSG\_HEADER\_LENGTH dos valores MaxMsgLength das filas de transmissão ao determinar o limite para BufferLength.

**Nota:** Somente a falha em obedecer à condição 1 pode ser diagnosticada de forma síncrona (com o código de razão MQRC\_MSG\_TOO\_BIG\_FOR\_Q ou MQRC\_MSG\_TOO\_BIG\_FOR\_Q\_MGR) quando a mensagem é colocada. Se as condições 2 ou 3 não forem satisfeitas, a mensagem será redirecionada para uma fila de mensagens não entregues (não entregues), em um gerenciador de filas intermediário ou no gerenciador de filas de destino. Se isso acontecer, uma mensagem de relatório será gerada se uma foi solicitada pelo emissor.

### de Saída

Tipo: MQBYTEExBufferComprimento-entrada

Este é um buffer que contém os dados do aplicativo a serem enviados. O buffer deve ser alinhado em um limite apropriado para a natureza dos dados na mensagem. O alinhamento de 4 bytes é adequado para a maioria das mensagens (incluindo mensagens que contêm estruturas de cabeçalho IBM MQ), mas algumas mensagens podem requerer alinhamento mais rigoroso. Por exemplo, uma mensagem contendo um número inteiro binário de 64 bits pode requerer alinhamento de 8 bytes.

Se Buffer contiver dados numéricos ou de caracteres, configure os campos CodedCharSetId e Encoding no parâmetro **MsgDesc** para os valores apropriados para os dados; isso permite que o receptor da mensagem converta os dados (se necessário) para o conjunto de caracteres e a codificação usados pelo receptor.

**Nota:** Todos os outros parâmetros na chamada MQPUT devem estar no conjunto de caracteres e na codificação do gerenciador de filas locais (fornecidos pelo atributo do gerenciador de filas **CodedCharSetId** e MQENC\_NATIVE).

Na linguagem de programação C, o parâmetro é declarado como um ponteiro para vazio; o endereço de qualquer tipo de dados pode ser especificado como o parâmetro.

Se o parâmetro **BufferLength** for zero, Buffer não será referido; nesse caso, o endereço do parâmetro transmitido por programas gravados em C ou no assembler System/390 poderá ser nulo...

### CompCode

Tipo: MQLONG - saída

O código de conclusão; é um dos seguintes:

#### **MQCC\_OK**

Indica conclusão bem-sucedida.

#### **MQCC\_WARNING**

Aviso (conclusão parcial).

#### **MQCC\_FAILED**

A chamada falhou.

### Razão

Tipo: MQLONG - saída

O código de razão qualificando CompCode.

Se CompCode for MQCC\_OK:

#### **MQRC\_NONE**

(0, X'000') Nenhuma razão a relatar.

Se CompCode for MQCC\_WARNING:

#### **MQRC\_INCOMPLETE\_GROUP**

(2241, X'8C1') Grupo de mensagens não concluído.

#### **MQRC\_INCOMPLETE\_MSG**

(2242, X'8C2') Mensagem lógica não concluída.

**MQRC\_INCONSISTENT\_PERSISTENCE**

(2185, X'889 ') Especificação de persistência inconsistente.

**MQRC\_INCONSISTENT\_UOW**

(2245, X'8C5') Especificação de unidade de trabalho inconsistente.

**MQRC\_MULTIPLE\_REASONS**

(2136, X'858 ') Códigos de razão múltiplos retornados.

**MQRC\_PRIORITY\_EXCEEDS\_MAXIMUM**

(2049, X'801 ') A prioridade da mensagem excede o valor máximo suportado.

**MQRC\_UNKNOWN\_REPORT\_OPTION**

(2104, X'838 ') Opções de relatório no descritor de mensagens não reconhecidas.

Se CompCode for MQCC\_FAILED:

**MQRC\_ADAPTER\_NOT\_AVAILABLE**

(2204, X'89C') Adaptador não disponível.

**MQRC\_ADAPTER\_SERV\_LOAD\_ERROR**

(2130, X'852') Não foi possível carregar o módulo de serviço do adaptador.

**MQRC\_ALIAS\_TARGTYPE\_CHANGED**

(2480, X'09B0') O tipo de destino de assinatura foi alterado de fila para objeto de tópico.

**MQRC\_API\_EXIT\_ERROR**

(2374, X'946') Falha na saída de API.

**MQRC\_API\_EXIT\_LOAD\_ERROR**

(2183, X'887') Não foi possível carregar a saída de API.

**MQRC\_ASID\_MISMATCH**

(2157, X'86D') ASIDs de início e primárias são diferentes.

**MQRC\_BACKED\_OUT**

(2003, X'7D3') Unidade de trabalho retornada.

**MQRC\_BUFFER\_ERROR**

(2004, X'7D4') Parâmetro de buffer inválido.

**MQRC\_BUFFER\_LENGTH\_ERROR**

(2005, X'7D5') Parâmetro de comprimento de buffer inválido.

**MQRC\_CALL\_IN\_PROGRESS**

(2219, X'8AB') Chamada MQI inserida antes da chamada anterior concluída.

**MQRC\_CALL\_INTERRUPTED**

(2549, X'9F5') MQPUT ou MQCMIT foi interrompido e o processamento de reconexão não pode restabelecer um resultado definitivo.

**MQRC\_CF\_NOT\_AVAILABLE**

(2345, X' 929 ') Recurso de acoplamento não disponível.

**MQRC\_CF\_STRUC\_FAILED**

(2373, X'945') Falha na estrutura do recurso de acoplamento.

**MQRC\_CF\_STRUC\_IN\_USE**

(2346, X'92A') Estrutura de recurso de acoplamento em uso.

**MQRC\_CFGR\_ERROR**

(2416, X' 970 ') A estrutura do parâmetro do grupo PCF MQCFGR nos dados da mensagem não é válida..

**MQRC\_CFH\_ERROR**

(2235, X'8BB') Estrutura do cabeçalho PCF não válida.

**MQRC\_CFIF\_ERROR**

(2414, X'96E') A estrutura do parâmetro de filtro inteiro PCF nos dados da mensagem não é válida.



**MQRC\_CFIL\_ERROR**

(2236, X'8BC') A estrutura do parâmetro da lista de inteiros PCF ou a estrutura do parâmetro da lista de inteiros PCIF\*64 não é válida.

**MQRC\_CFIN\_ERROR**

(2237, X'8BD') Estrutura de parâmetro de número inteiro PCF ou estrutura de parâmetro de número inteiro PCIF\*64 não é válida.

**MQRC\_CFSF\_ERROR**

(2415, X'96F') A estrutura do parâmetro de filtro de sequência PCF nos dados da mensagem não é válida..

**MQRC\_CFSL\_ERROR**

(2238, X'8BE') Estrutura do parâmetro da lista de cadeias PCF não é válida.

**MQRC\_CFST\_ERROR**

(2239, X'8BF') Estrutura do parâmetro de sequência PCF não válida.

**MQRC\_CICS\_WAIT\_FAILED**

(2140, X'85C') Pedido de espera rejeitado por CICS.

**MQRC\_CLUSTER\_EXIT\_ERROR**

(2266, X'8DA') Falha na saída de carga de trabalho do cluster.

**MQRC\_CLUSTER\_RESOLUTION\_ERROR**

(2189, X'88D') Falha na resolução do nome do cluster.

**MQRC\_CLUSTER\_RESOURCE\_ERROR**

(2269, X'8DD') Erro de recurso de cluster

**MQRC\_COD\_NOT\_VALID\_FOR\_XCF\_Q**

(2106, X'83A') A opção de relatório COD não é válida para a fila XCF.

**MQRC\_CONNECTION\_BROKEN**

(2009, X'7D9') Conexão ao gerenciador de filas perdida.

**MQRC\_CONNECTION\_NOT\_AUTHORIZED**

(2217, X'8A9') Não autorizado para conexão.

**MQRC\_CONNECTION QUIESCING**

(2202, X'89A') Quiesce de conexão.

**MQRC\_CONNECTION\_STOPPING**

(2203, X'89B') Conexão sendo encerrada.

**MQRC\_CONTENT\_ERROR**

2554 (X'09FA') O conteúdo da mensagem não pôde ser analisado para determinar se a mensagem deve ser entregue a um assinante com um seletor de mensagem estendida.

**MQRC\_CONTEXT\_HANDLE\_ERROR**

(2097, X'831 ') O identificador de fila referido não salva o contexto..

**MQRC\_CONTEXT\_NOT\_AVAILABLE**

(2098, X'832 ') Contexto não disponível para o identificador de fila referido.

**MQRC\_DATA\_LENGTH\_ERROR**

(2010, X'7DA') Parâmetro de comprimento de dados inválido.

**MQRC\_DH\_ERROR**

(2135, X'857 ') Estrutura do cabeçalho de distribuição inválida.

**MQRC\_DLH\_ERROR**

(2141, X'85D') Estrutura do cabeçalho de devoluções não válida.

**MQRC\_EPH\_ERROR**

(2420, X' 974 ') A estrutura PCF integrada não é válida.

**MQRC\_EXPIRY\_ERROR**

(2013, X'7DD') Tempo de expiração inválido.

**MQRC\_FEEDBACK\_ERROR**

(2014, X'7DE') Código de feedback inválido.

**MQRC\_GLOBAL\_UOW\_CONFLICT**  
(2351, X'92F') Unidades de trabalho globais conflitam.

**MQRC\_GROUP\_ID\_ERROR**  
(2258, X'8D2') Identificador de grupo inválido.

**MQRC\_HANDLE\_IN\_USE\_FOR\_UOW**  
(2353, X'931') Manipulação em uso para unidade de trabalho global.

**MQRC\_HCONN\_ERROR**  
(2018, X'7E2') Manipulação de conexões não válida.

**MQRC\_HEADER\_ERROR**  
(2142, X'85E') Estrutura do cabeçalho MQ não válida.

**MQRC\_HOBJ\_ERROR**  
(2019, X'7E3') Manipulação de objetos não válida.

**MQRC\_IIH\_ERROR**  
(2148, X'864 ') IMS estrutura do cabeçalho de informações não válida.

**MQRC\_INCOMPLETE\_GROUP**  
(2241, X'8C1') Grupo de mensagens não concluído.

**MQRC\_INCOMPLETE\_MSG**  
(2242, X'8C2') Mensagem lógica não concluída.

**MQRC\_INCONSISTENT\_PERSISTENCE**  
(2185, X'889 ') Especificação de persistência inconsistente.

**MQRC\_INCONSISTENT\_UOW**  
(2245, X'8C5') Especificação de unidade de trabalho inconsistente.

**MQRC\_LOCAL\_UOW\_CONFLICT**  
(2352, X'930') Unidade de trabalho global conflita com unidade de trabalho local.

**MQRC\_MD\_ERROR**  
(2026, X'7EA') Descritor de mensagem inválido.

**MQRC\_MDE\_ERROR**  
(2248, X'8C8') Extensão do descritor de mensagens inválida.

**MQRC\_MISSING\_REPLY\_TO\_Q**  
(2027, X'7EB') Fila de resposta ausente ou MQPMO\_SUPPRESS\_REPLYTO foi usado

**MQRC\_MISSING\_WIH**  
(2332, X'91C') Os dados da mensagem não começam com MQWIH.

**MQRC\_MSG\_FLAGS\_ERROR**  
(2249, X'8C9') Sinalizadores de mensagem não válidos.

**MQRC\_MSG\_SEQ\_NUMBER\_ERROR**  
(2250, X'8CA') Número de sequência da mensagem inválido.

**MQRC\_MSG\_TOO\_BIG\_FOR\_Q**  
(2030, X'7EE') Comprimento da mensagem maior que o máximo para a fila.

**MQRC\_MSG\_TOO\_BIG\_FOR\_Q\_MGR**  
(2031, X'7EF') Comprimento da mensagem maior que o máximo para o gerenciador de filas.

**MQRC\_MSG\_TYPE\_ERROR**  
(2029, X'7ED') Tipo de mensagem no descritor de mensagens inválido.

**MQRC\_MULTIPLE\_REASONS**  
(2136, X'858 ') Códigos de razão múltiplos retornados.

**MQRC\_NO\_DESTINATIONS\_AVAILABLE**  
(2270, X'8DE') Nenhuma fila de destinos disponível.

**MQRC\_NOT\_OPEN\_FOR\_OUTPUT**  
(2039, X'7F7') Fila não aberta para saída.

**MQRC\_NOT\_OPEN\_FOR\_PASS\_ALL**  
(2093, X'82D') Fila não aberta para passar todo o contexto.

**MQRC\_NOT\_OPEN\_FOR\_PASS\_IDENT**

(2094, X'82E') Fila não aberta para transmitir contexto de identidade.

**MQRC\_NOT\_OPEN\_FOR\_SET\_ALL**

(2095, X'82F') Fila não aberta para configurar todo o contexto.

**MQRC\_NOT\_OPEN\_FOR\_SET\_IDENT**

(2096, X'830 ') Fila não aberta para o contexto de identidade configurado.

**MQRC\_OBJECT\_CHANGED**

(2041, X'7F9') Definição de objeto alterada desde a abertura.

**MQRC\_OBJECT\_DAMAGED**

(2101, X'835') Objeto danificado.

**MQRC\_OFFSET\_ERROR**

(2251, X'8CB') O deslocamento do segmento da mensagem não é válido..

**MQRC\_OPEN\_FAILED**

(2137, X'859') Objeto não aberto com sucesso.

**MQRC\_OPTIONS\_ERROR**

(2046, X'7FE') Opções não válidas ou não consistentes.

**MQRC\_ORIGINAL\_LENGTH\_ERROR**

(2252, X'8CC') Comprimento original inválido.

**MQRC\_PAGESET\_ERROR**

(2193, X'891') Erro ao acessar o conjunto de dados do conjunto de páginas.

**MQRC\_PAGESET\_FULL**

(2192, X'890') O meio de armazenamento externo está cheio.

**MQRC\_PCF\_ERROR**

(2149, X'865 ') estruturas PCF inválidas.

**MQRC\_PERSISTENCE\_ERROR**

(2047, X'7FF') Persistência inválida.

**MQRC\_PERSISTENT\_NOT\_ALLOWED**

(2048, X'800 ') A Fila não suporta mensagens persistentes.

**MQRC\_PMO\_ERROR**

(2173, X'87D') Estrutura de opções de mensagem de entrada inválida.

**MQRC\_PMO\_RECORD\_FLAGS\_ERROR**

(2158, X'86E') Sinalizadores de registro de mensagem de colocação não válidos.

**MQRC\_PRIORITY\_ERROR**

(2050, X'802 ') A prioridade da mensagem não é válida.

**MQRC\_PUBLICATION\_FAILURE**

(2502, X'9C6') A publicação não foi entregue a nenhum dos assinantes.

**MQRC\_PUT\_INHIBITED**

(2051, X'803 ') Chamadas de Put inibidas para a fila, para a fila para a qual essa fila é resolvida ou o tópico

**MQRC\_PUT\_MSG\_RECORDS\_ERROR**

(2159, X'86F') Os registros de mensagens Put não são válidos.

**MQRC\_PUT\_NOT\_RETIDO**

(2479, X'09AF') A publicação não pôde ser retida.

**MQRC\_Q\_DELETED**

(2052, X'804') A fila foi excluída.

**MQRC\_Q\_FULL**

(2053, X'805 ') A fila já contém número máximo de mensagens.

**MQRC\_Q\_MGR\_NAME\_ERROR**

(2058, X'80A') Nome do gerenciador de filas não válido ou não conhecido.

**MQRC\_Q\_MGR\_NOT\_AVAILABLE**

(2059, X'80B') Gerenciador de filas não disponível para conexão.

**MQRC\_Q\_MGR QUIESCING**

(2161, X'871') Quiesce do gerenciador de filas.

**MQRC\_Q\_MGR\_STOPPING**

(2162, X'872') Gerenciador de filas sendo encerrado.

**MQRC\_Q\_SPACE\_NOT\_AVAILABLE**

(2056, X'808 ') Nenhum espaço disponível no disco para fila.

**MQRC\_RECONNECT\_FAILED**

(2548, X'9F4') Após a reconexão, ocorreu um erro ao restabelecer os identificadores para uma conexão reconectável...

**MQRC\_RECS\_PRESENT\_ERROR**

(2154, X'86A') Número de registros presentes não válidos.

**MQRC\_REPORT\_OPTIONS\_ERROR**

(2061, X'80D') Opções de relatório no descritor de mensagem não válidas.

**MQRC\_RESOURCE\_PROBLEM**

(2102, X'836') Recursos insuficientes do sistema disponíveis.

**MQRC\_RESPONSE\_RECORDS\_ERROR**

(2156, X'86C') Registros de resposta inválidos.

**MQRC\_RFH\_ERROR**

(2334, X'91E') A estrutura MQRFH ou MQRFH2 não é válida.

**MQRC\_RMH\_ERROR**

(2220, X'8AC') A estrutura do cabeçalho da mensagem de referência não é válida.

**MQRC\_SEGMENT\_LENGTH\_ZERO**

(2253, X'8CD') O comprimento dos dados no segmento de mensagem é zero...

**MQRC\_SEGMENTS\_NOT\_SUPPORTED**

(2365, X'93D') Segmentos não suportados.

**MQRC\_SELECTION\_NOT\_AVAILABLE**

2551 (X'09F7') Existe um possível assinante para a publicação, mas o gerenciador de filas não pode verificar se a publicação deve ser enviada para o assinante.

**MQRC\_STOPPED\_BY\_CLUSTER\_EXIT**

(2188, X'88C') Chamada rejeitada pela saída de carga de trabalho do cluster.

**MQRC\_STORAGE\_CLASS\_ERROR**

(2105, X'839 ') Erro da classe de armazenamento..

**MQRC\_STORAGE\_MEDIUM\_FULL**

(2192, X'890') O meio de armazenamento externo está cheio.

**MQRC\_STORAGE\_NOT\_AVAILABLE**

(2071, X'817') Armazenamento insuficiente disponível.

**MQRC\_SUPPRESSED\_BY\_EXIT**

(2109, X'83D') Chamada suprimida pelo programa de saída.

**MQRC\_SYNCPOINT\_LIMIT\_REACHED**

(2024, X'7E8') Nenhuma mensagem adicional pode ser manipulada dentro da unidade de trabalho atual.

**MQRC\_SYNCPOINT\_NOT\_AVAILABLE**

(2072, X'818 ') Suporte de ponto de sincronização não disponível.

**MQRC\_TM\_ERROR**

(2265, X'8D9') Estrutura da mensagem do acionador inválida.

**MQRC\_TMC\_ERROR**

(2191, X'88F') A estrutura da mensagem do acionador de caracteres não é válida..

**MQRC\_UNEXPECTED\_ERROR**

(2195, X'893') Ocorreu um erro inesperado.

**MQRC\_UOW\_ENLISTMENT\_ERROR**

(2354, X'932') Cadastramento na unidade de trabalho global falhou.

**MQRC\_UOW\_MIX\_NOT\_SUPPORTED**

(2355, X'933') Combinação de chamadas de unidade de trabalho não suportada.

**MQRC\_UOW\_NOT\_AVAILABLE**

(2255, X'8CF') Unidade de trabalho não disponível para o gerenciador de filas a usar.

**MQRC\_WIH\_ERROR**

(2333, X'91D') Estrutura MQWIH não válida.

**MQRC\_WRONG\_MD\_VERSION**

(2257, X'8D1') Versão errada do MQMD fornecido.

**MQRC\_XQH\_ERROR**

(2260, X'8D4') A estrutura do cabeçalho da fila de transmissão não é válida.

Para obter informações detalhadas sobre esses códigos, consulte [Mensagens e códigos de razão](#).

## Notas de uso do tópico

1. As notas a seguir se aplicam ao uso de tópicos:

a. Ao usar MQPUT para publicar mensagens em um tópico, em que um ou mais assinantes para esse tópico não podem receber a publicação devido a um problema com sua fila de assinantes (por exemplo, ela está cheia), o código de Razão retornado para a chamada MQPUT e o comportamento de entrega depende da configuração dos atributos PMSGDLV ou NPMSGDLV no TOPIC. A entrega de uma publicação para a fila de mensagens não entregues quando MQRO\_DEAD\_LETTER\_Q é especificado ou o descarte da mensagem quando MQRO\_DISCARD\_MSG é especificado é considerado como uma entrega bem-sucedida da mensagem. Se nenhuma das publicações for entregue, o MQPUT retornará com MQRC\_PUBLICATION\_FAILURE Isso pode ocorrer nos casos a seguir:

- Uma mensagem é publicada para um TOPIC com PMSGDLV ou NPMSGDLV (dependendo da persistência da mensagem) configurado como ALL e qualquer assinatura (durável ou não) tem uma fila que não pode receber a publicação..
- Uma mensagem é publicada para um TOPIC com PMSGDLV ou NPMSGDLV (dependendo da persistência da mensagem) configurado como ALLDUR e uma assinatura durável possui uma fila que não pode receber a publicação

O MQPUT pode retornar com MQRC\_NONE, embora as publicações não tenham sido entregues a alguns assinantes nos casos a seguir:

- Uma mensagem é publicada para um TOPIC com PMSGDLV ou NPMSGDLV (dependendo da persistência da mensagem) definido para ALLAVAIL e qualquer assinatura, durável ou não, tem uma fila que não pode receber a publicação.
- Uma mensagem é publicada para um TOPIC com PMSGDLV ou NPMSGDLV (dependendo da persistência da mensagem) configurado para ALLDUR e uma assinatura não durável possui uma fila que não pode receber a publicação.

É possível usar o atributo de tópico USEDLQ para determinar se a fila de mensagens não entregues é usada quando as mensagens de publicação não podem ser entregues para a fila de assinantes correta. Para obter mais informações sobre o uso de USEDLQ, consulte [DEFINE TOPIC](#)

b. Se não houver assinantes para o tópico sendo usado, a mensagem publicada não será enviada para nenhuma fila e será descartada. Não importa se a mensagem é persistente ou não persistente, ou se tem expiração ilimitada ou tem um tempo de expiração, ela ainda será descartada se não houver assinantes. A exceção a isso é se a mensagem deve ser retida, nesse caso, embora não seja enviada para nenhuma fila de assinantes, ela é armazenada com relação ao tópico a ser entregue para quaisquer novas assinaturas ou para quaisquer assinantes que solicitam publicações retidas usando MQSUBRQ.

## MQPUT e MQPUT1

É possível usar as chamadas MQPUT e MQPUT1 para colocar mensagens em uma fila; a chamada a ser usada depende das circunstâncias

- Use a chamada MQPUT para colocar várias mensagens na mesma fila..

Uma chamada MQOPEN especificando a opção MQOO\_OUTPUT é emitida primeiro, seguida por uma ou mais solicitações MQPUT para incluir mensagens na fila; finalmente, a fila é fechada com uma chamada MQCLOSE. Isso fornece melhor desempenho do que o uso repetido da chamada MQPUT1 ..

- Use a chamada MQPUT1 para colocar apenas uma mensagem em uma fila..

Essa chamada encapsula as chamadas MQOPEN, MQPUT e MQCLOSE em uma única chamada, minimizando o número de chamadas que devem ser emitidas..

## Filas de destino

As notas a seguir se aplicam ao uso de filas de destino:

1. Se um aplicativo colocar uma sequência de mensagens na mesma fila sem usar grupos de mensagens, A ordem dessas mensagens é preservada se as condições detalhadas forem satisfeitas; Algumas condições se aplicam às filas de destino locais e remotas; outras condições se aplicam apenas às filas de destino remotas


### Condições que se aplicam a filas de destino locais e remotas

- Todas as chamadas MQPUT estão na mesma unidade de trabalho ou nenhuma delas está dentro de uma unidade de trabalho.


Esteja ciente de que quando as mensagens são colocadas em uma fila específica em uma única unidade de trabalho, as mensagens de outros aplicativos podem ser intercaladas com a sequência de mensagens na fila.

- Todas as chamadas MQPUT são feitas usando o mesmo identificador de objeto *Hobj*..

Em alguns ambientes, a sequência de mensagens também é preservada quando diferentes identificadores de objetos são usados, se as chamadas forem feitas a partir do mesmo aplicativo. O significado de *mesmo aplicativo* é determinado pelo ambiente:

–  No z/OS, o aplicativo é:.

- Para CICS, a tarefa CICS
- Para IMS, a tarefa
- Para o lote z/OS , a tarefa

–  No IBM i, o aplicativo é a tarefa

–  No AIX, Linux, and Windows, o aplicativo é o encadeamento

- Todas as mensagens tenham a mesma prioridade.
- As mensagens não são colocadas em uma fila de cluster com MQOO\_BIND\_NOT\_FIXED especificado (ou com MQOO\_BIND\_AS\_Q\_DEF em vigor quando o atributo da fila DefBind possui o valor MQBND\_BIND\_NOT\_FIXED)....

### Condições adicionais que se aplicam às filas de destino remotas

- Há apenas um caminho do gerenciador da fila de envio para o gerenciador de filas de destino

Se algumas mensagens na sequência puderem seguir um caminho diferente (por exemplo, devido à reconfiguração, ao balanceamento de tráfego ou à seleção de caminho com base no tamanho da mensagem), a ordem das mensagens no gerenciador de filas de destino não poderá ser garantida.

- As mensagens não são colocadas temporariamente em filas de mensagens não entregues nos gerenciadores de filas de envio, intermediário ou de destino

Se uma ou mais das mensagens forem colocadas temporariamente em uma fila de mensagens não entregues (por exemplo, porque uma fila de transmissão ou a fila de destino está temporariamente cheia), as mensagens poderão chegar na fila de destino fora da sequência...

- As mensagens são persistentes ou não persistentes.

Se um canal na rota entre os gerenciadores de filas de envio e de destino tiver seu atributo **NonPersistentMsgSpeed** configurado como MQNPMS\_FAST, as mensagens não persistentes poderão saltar à frente das mensagens persistentes, resultando na ordem das mensagens persistentes relativas às mensagens não persistentes não sendo preservadas. No entanto, a ordem de mensagens persistentes relativas entre si e de mensagens não persistentes relativas entre si é preservada.

Se essas condições não forem satisfeitas, será possível usar grupos de mensagens para preservar a ordem da mensagem, mas isso requer que ambos os aplicativos de envio e de recebimento usem o suporte de agrupamento de mensagem. Para obter informações adicionais sobre grupos de mensagens, consulte:

- [MQMD - Campo MsgFlags](#)
- [MQPMO\\_LOGICAL\\_ORDER](#)
- [MQGMO\\_LOGICAL\\_ORDER](#)

## Listas de Distribuição

As seguintes notas aplicam-se ao uso de listas de distribuição.

As listas de distribuição são suportadas nos seguintes ambientes:

-  AIX
-  IBM i
-  Linux
-  Windows

e para o IBM MQ MQI clients conectado a esses sistemas

1. É possível colocar mensagens em uma lista de distribuição usando uma version-1 ou uma version-2 MQPMO. Se você usar um MQPMO version-1 (ou um MQPMO version-2 com RecsPresent igual a zero), o aplicativo poderá fornecer nenhum registro de mensagem de colocação ou registro de resposta. Não é possível identificar as filas que encontram erros se a mensagem for enviada com êxito para algumas filas na lista de distribuição e não para outras.

Se o aplicativo fornecer registros de mensagens put ou registros de resposta, configure o campo Version para MQPMO\_VERSION\_2.

Também é possível usar um MQPMO version-2 para enviar mensagens para uma única fila que não esteja em uma lista de distribuição, assegurando que RecsPresent seja zero.

2. Os parâmetros de código de conclusão e de código de razão são configurados conforme a seguir::

- Se todas as colocações nas filas na lista de distribuição forem bem-sucedidas ou falharem da mesma maneira, os parâmetros do código de conclusão e do código de razão serão configurados para descrever o resultado comum. Os registros de resposta MQRR (se fornecidos pelo aplicativo) não são configurados neste caso.

Por exemplo, se cada put for bem-sucedido, o código de finalização e o código de razão serão configurados como MQCC\_OK e MQRC\_NONE; se cada put falhar porque todas as filas são inibidas para puts, os parâmetros serão configurados como MQCC\_FAILED e MQRC\_PUT\_INIBITED

- Se as colocações nas filas na lista de distribuição não forem bem-sucedidas ou falharem da mesma maneira:

- O parâmetro de código de conclusão será configurado como MQCC\_WARNING se pelo menos um put for bem-sucedido e como MQCC\_FAILED se todos falharem.
- O parâmetro do código de razão é configurado como MQRC\_MULTIPLE\_REASON
- Os registros de resposta (se fornecidos pelo aplicativo) são configurados para os códigos de conclusão individuais e os códigos de razão para as filas na lista de distribuição

Se o put para um destino falhar porque a abertura para esse destino falhou, os campos no registro de resposta serão configurados para MQCC\_FAILED e MQRC\_OPEN\_FAILED; esse destino será incluído no InvalidDestCount

3. Se um destino na lista de distribuição for resolvido para uma fila local, a mensagem será colocada nessa fila no formato normal (ou seja, não como uma mensagem da lista de distribuições). Se mais de um destino for resolvido para a mesma fila local, uma mensagem será colocada na fila para cada destino.

Se um destino na lista de distribuição for resolvido para uma fila remota, uma mensagem será colocada na fila de transmissão apropriada. Quando vários destinos forem resolvidos para a mesma fila de transmissão, uma única mensagem da lista de distribuições contendo esses destinos poderá ser colocada na fila de transmissão, mesmo que esses destinos não fossem adjacentes na lista de destinos fornecidos pelo aplicativo. No entanto, isso poderá ser feito apenas se a fila de transmissão suportar mensagens da lista de distribuição (consulte [DistLists](#)).

Se a fila de transmissão não suportar listas de distribuição, uma cópia da mensagem no formato normal será colocada na fila de transmissão para cada destino que usa essa fila de transmissão..

Se uma lista de distribuição com os dados da mensagem do aplicativo for muito grande para uma fila de transmissão, a mensagem da lista de distribuição será dividida em mensagens da lista de distribuição menores, cada uma contendo menos destinos. Se os dados da mensagem do aplicativo apenas se ajustarem na fila, as mensagens da lista de distribuições não poderão ser usadas e o gerenciador de filas gerará uma cópia da mensagem no formato normal para cada destino que usar essa fila de transmissão.

Se destinos diferentes tiverem prioridade ou persistência de mensagem diferente (isso pode ocorrer quando o aplicativo especificar MQPRI\_PRIORITY\_AS\_Q\_DEF ou MQPER\_PERSISTENCE\_AS\_Q\_DEF), as mensagens não serão mantidas na mesma mensagem da lista de distribuição. Em vez disso, o gerenciador de filas gera quantas mensagens de lista de distribuição forem necessárias para acomodar os diferentes valores de prioridade e persistência.

4. Uma colocação em uma lista de distribuição pode resultar em:

- Uma única mensagem de lista de distribuição ou
- Um número de mensagens menores da lista de distribuição ou
- Uma mistura de mensagens da lista de distribuições e mensagens normais ou
- Apenas mensagens normais..

Qual dos itens acima ocorre depende se:

- Os destinos na lista são locais, remotos ou uma mistura.
- Os destinos têm a mesma prioridade e persistência de mensagem.
- As filas de transmissão podem conter mensagens da lista de distribuições
- Os comprimentos máximos de mensagens das filas de transmissão são suficientemente grandes para acomodar a mensagem em forma de lista de distribuições.

No entanto, independentemente de qual das ocorrências acima, cada mensagem *física* resultante (ou seja, cada mensagem normal ou mensagem da lista de distribuição resultante da colocação) conta como apenas *uma* mensagem quando:

- Verificando se o aplicativo excedeu o número máximo permitido de mensagens em uma unidade de trabalho (consulte o atributo do gerenciador de filas **MaxUncommittedMsgs**).
- Verificando se as condições de acionamento foram atendidas



- Incrementar as profundidades da fila e verificar se a profundidade máxima da fila das filas seria excedida.
5. Qualquer mudança nas definições de fila que teria feito com que uma manipulação se tornasse inválida se as filas fossem abertas individualmente (por exemplo, uma mudança no caminho de resolução) não faz com que a manipulação da lista de distribuição se torne inválida. No entanto, isso resulta em uma falha para essa fila específica quando a manipulação da lista de distribuições é usada em uma chamada MQPUT subsequente

## Cabeçalhos

Se uma mensagem for colocada com uma ou mais estruturas de cabeçalho do IBM MQ no início dos dados da mensagem do aplicativo, o gerenciador de filas executará determinadas verificações nas estruturas de cabeçalho para verificar se elas são válidas. Se o gerenciador de filas detectar um erro, a chamada falhará com um código de razão apropriado. As verificações realizadas variam de acordo com as estruturas específicas que estão presentes:

- As verificações serão executadas apenas se um MQMD version-2 ou posterior for usado na chamada MQPUT ou MQPUT1. As verificações não serão executadas se um MQMD version-1 for usado, mesmo se um MQMDE estiver presente no início dos dados da mensagem.
- As estruturas que não são suportadas pelo gerenciador de filas locais e as estruturas após o primeiro MQDLH na mensagem não são validadas.
- As estruturas MQDH e MQMDE são validadas completamente pelo gerenciador de filas..
- Outras estruturas são validadas parcialmente pelo gerenciador de filas (nem todos os campos são verificados)

As verificações gerais executadas pelo gerenciador de filas incluem o seguinte:

- O campo `StrucId` deve ser válido..
- O campo `Version` deve ser válido..
- O campo `StrucLength` deve especificar um valor que seja grande o suficiente para incluir a estrutura mais quaisquer dados de comprimento variável que façam parte da estrutura.
- O campo `CodedCharSetId` não deve ser zero ou um valor negativo que não seja válido (MQCCSI\_DEFAULT, MQCCSI\_EMBEDDED, MQCCSI\_Q\_MGR e MQCCSI\_UNDEFINED não são válidos na maioria das estruturas do cabeçalho IBM MQ).
- O parâmetro **BufferLength** da chamada deve especificar um valor que seja grande o suficiente para incluir a estrutura (a estrutura não deve estender além do final da mensagem).

Para além do controlo geral das estruturas, devem ser satisfeitas as seguintes condições:

- A soma dos comprimentos das estruturas em uma mensagem PCF deve ser igual ao comprimento especificado pelo parâmetro **BufferLength** na chamada MQPUT ou MQPUT1. Uma mensagem PCF é uma mensagem que possui um nome de formato MQFMT\_ADMIN, MQFMT\_EVENT, ou MQFMT\_PCF.
- Uma estrutura IBM MQ não deve ser truncada, exceto nas seguintes situações em que estruturas truncadas são permitidas:
  - As mensagens que são mensagens de relatório
  - Mensagens PCF..
  - Mensagens contendo uma estrutura MQDLH. (As estruturas após o primeiro MQDLH podem ser truncadas; as estruturas que precedem o MQDLH não podem.)
- Uma estrutura IBM MQ não deve ser dividida em dois ou mais segmentos; a estrutura deve estar inteiramente contida dentro de um segmento

## de Saída

Para a linguagem de programação Visual Basic, os seguintes pontos se aplicam:

- Se o tamanho do parâmetro **Buffer** for menor que o comprimento especificado pelo parâmetro **BufferLength**, a chamada falhará com código de razão MQRC\_BUFFER\_LENGTH\_ERROR.
- O parâmetro **Buffer** é declarado como sendo do tipo String. Se os dados a serem colocados na fila não forem do tipo String, use a chamada MQPUTAny no lugar de MQPUT.

A chamada MQPUTAny tem os mesmos parâmetros que a chamada MQPUT, exceto que o parâmetro **Buffer** é declarado como sendo do tipo Any, permitindo que qualquer tipo de dados seja colocado na fila. No entanto, isso significa que Buffer não pode ser verificado para assegurar que tenha pelo menos BufferLength bytes de tamanho.

## Chamada C

```
MQPUT (Hconn, Hobj, &MsgDesc, &PutMsgOpts, BufferLength, Buffer,
       &CompCode, &Reason);
```

Declare os parâmetros conforme a seguir:

```
MQHCONN  Hconn;           /* Connection handle */
MQHOBJ   Hobj;           /* Object handle */
MQMD     MsgDesc;       /* Message descriptor */
MQPMO    PutMsgOpts;    /* Options that control the action of MQPUT */
MQLONG   BufferLength;   /* Length of the message in Buffer */
MQBYTE   Buffer[n];     /* Message data */
MQLONG   CompCode;     /* Completion code */
MQLONG   Reason;       /* Reason code qualifying CompCode */
```

## Chamada COBOL

```
CALL 'MQPUT' USING HCONN, HOBJ, MSGDESC, PUTMSGOPTS, BUFFERLENGTH,
                  BUFFER, COMPCODE, REASON.
```

Declare os parâmetros conforme a seguir:

```
** Connection handle
01 HCONN          PIC S9(9) BINARY.
** Object handle
01 HOBJ          PIC S9(9) BINARY.
** Message descriptor
01 MSGDESC.
   COPY CMQMDV.
** Options that control the action of MQPUT
01 PUTMSGOPTS.
   COPY CMQPMOV.
** Length of the message in BUFFER
01 BUFFERLENGTH PIC S9(9) BINARY.
** Message data
01 BUFFER       PIC X(n).
** Completion code
01 COMPCODE     PIC S9(9) BINARY.
** Reason code qualifying COMPCODE
01 REASON      PIC S9(9) BINARY.
```

## Chamada PL/I

```
call MQPUT (Hconn, Hobj, MsgDesc, PutMsgOpts, BufferLength, Buffer,
            CompCode, Reason);
```

Declare os parâmetros conforme a seguir:

```
dcl Hconn      fixed bin(31); /* Connection handle */
dcl Hobj      fixed bin(31); /* Object handle */
dcl MsgDesc    like MQMD;    /* Message descriptor */
```

```

dcl PutMsgOpts    like MQPMO;      /* Options that control the action of
                                   MQPUT */
dcl BufferLength  fixed bin(31);   /* Length of the message in Buffer */
dcl Buffer        char(n);        /* Message data */
dcl CompCode     fixed bin(31);   /* Completion code */
dcl Reason       fixed bin(31);   /* Reason code qualifying CompCode */

```

## Chamada do Assembler de Alto Nível

```

CALL MQPUT, (HCONN,HOBJ,MSGDESC,PUTMSGOPTS,BUFFERLENGTH, X
            BUFFER,COMPCODE,REASON)

```

Declare os parâmetros conforme a seguir:

HCONN	DS	F	Connection handle
HOBJ	DS	F	Object handle
MSGDESC	CMQMDA	,	Message descriptor
PUTMSGOPTS	CMQPMOA	,	Options that control the action of MQPUT
BUFFERLENGTH	DS	F	Length of the message in BUFFER
BUFFER	DS	CL(n)	Message data
COMPCODE	DS	F	Completion code
REASON	DS	F	Reason code qualifying COMPCODE

## Chamada do Visual Basic

Windows

```

MQPUT Hconn, Hobj, MsgDesc, PutMsgOpts, BufferLength, Buffer, CompCode,
      Reason

```

Declare os parâmetros conforme a seguir:

Dim Hconn	As Long	'Connection handle'
Dim Hobj	As Long	'Object handle'
Dim MsgDesc	As MQMD	'Message descriptor'
Dim PutMsgOpts	As MQPMO	'Options that control the action of MQPUT'
Dim BufferLength	As Long	'Length of the message in Buffer'
Dim Buffer	As String	'Message data'
Dim CompCode	As Long	'Completion code'
Dim Reason	As Long	'Reason code qualifying CompCode'

## MQPUT1 - Colocar uma mensagem

A chamada MQPUT1 coloca uma mensagem em uma fila, lista de distribuição ou em um tópico.

A fila, lista de distribuição ou tópico não precisa ser aberto.

### Sintaxe


MQPUT1 (*Hconn, ObjDesc, MsgDesc, PutMsgOpts, BufferLength, Buffer, CompCode, Motivo*)

### Parâmetros

#### Hconn

Tipo: MQHCONN - entrada

Esta manipulação representa a conexão ao gerenciador de filas. O valor de *Hconn* foi retornado por uma chamada anterior MQCONN ou MQCONNX.

 Em z/OS para aplicativos CICS, a chamada MQCONN pode ser omitida e o seguinte valor especificado para *Hconn*:

## **MQHC\_DEF\_HCONN**

Manipulação de conexões padrão.

### **ObjDesc**

Tipo: MQOD-entrada / saída

Esta é uma estrutura que identifica a fila na qual a mensagem é incluída, ou o tópico no qual a mensagem é publicada Consulte [“MQOD-Descrição de objetos”](#) na página 497 para obter detalhes.

Se a estrutura for uma fila, o usuário deverá estar autorizado a abrir a fila para saída A fila não deve ser modelo.

### **MsgDesc**

Tipo: MQMD - entrada/saída

Essa estrutura descreve os atributos da mensagem que está sendo enviada e recebe informações de feedback após a conclusão da solicitação de colocação Consulte [“MQMD - descritor de mensagem”](#) na página 435 para obter detalhes.

Se o aplicativo fornecer um MQMD version-1 , os dados da mensagem poderão ser prefixados com uma estrutura MQMDE para especificar valores para os campos existentes no MQMD version-2 , mas não no version-1. Configure o campo Format no MQMD para MQFMT\_MD\_EXTENSION para indicar que um MQMDE está presente Consulte [“MQMDE-Extensão do descritor de mensagens”](#) na página 488 para obter mais detalhes.

O aplicativo não precisa fornecer uma estrutura MQMD se uma manipulação de mensagem válida for fornecida no campo MsgHandle da estrutura MQGMO ou nos campos OriginalMsgHandle ou NewMsgHandle da estrutura MQPMO.. Se nada for fornecido em um desses campos, o descritor da mensagem será obtido do descritor associado aos identificadores de mensagens..

### **PutMsgOpts**

Tipo: MQPMO-entrada / saída

Consulte [“MQPMO-Opções de mensagem de colocação”](#) na página 518 para obter detalhes.

### **BufferLength**

Tipo: MQLONG - entrada

O comprimento da mensagem em Buffer Zero é válido e indica que a mensagem não contém nenhum dado do aplicativo. O limite superior depende de vários fatores; consulte [“MQPUT-Mensagem de colocação”](#) na página 777 para obter a descrição do parâmetro **BufferLength**

### **de Saída**

Tipo: MQBYTEExBufferComprimento-entrada

Este é um buffer que contém os dados da mensagem do aplicativo a serem enviados Alinhe o buffer em um limite apropriado para a natureza dos dados na mensagem. O alinhamento de 4 bytes é adequado para a maioria das mensagens (incluindo mensagens que contêm estruturas de cabeçalho IBM MQ ), mas algumas mensagens podem requerer alinhamento mais rigoroso. Por exemplo, uma mensagem contendo um número inteiro binário de 64 bits pode requerer alinhamento de 8 bytes.

Se Buffer contiver dados numéricos ou de caracteres, configure os campos CodedCharSetId e Encoding no parâmetro **MsgDesc** para os valores apropriados para os dados; isso permite que o receptor da mensagem converta os dados (se necessário) para o conjunto de caracteres e a codificação usados pelo receptor.

**Nota:** Todos os outros parâmetros na chamada MQPUT1 devem estar no conjunto de caracteres e na codificação do gerenciador de fila local (fornecido pelo atributo do gerenciador de filas **CodedCharSetId** e MQENC\_NATIVE).

Na linguagem de programação C, o parâmetro é declarado como um ponteiro para vazio; o endereço de qualquer tipo de dados pode ser especificado como o parâmetro.

Se o parâmetro **BufferLength** for zero, Buffer não será referido; nesse caso, o endereço do parâmetro transmitido por programas gravados em C ou no assembler System/390 poderá ser nulo...

**CompCode**

Tipo: MQLONG - saída

O código de conclusão; é um dos seguintes:

**MQCC\_OK**

Indica conclusão bem-sucedida.

**MQCC\_WARNING**

Aviso (conclusão parcial).

**MQCC\_FAILED**

A chamada falhou.

**Razão**

Tipo: MQLONG - saída

O código de razão qualificando CompCode.

Se CompCode for MQCC\_OK:

**MQRC\_NONE**

(0, X'000') Nenhuma razão a relatar.

Se CompCode for MQCC\_WARNING:

**MQRC\_MULTIPLE\_REASONS**

(2136, X'858 ') Códigos de razão múltiplos retornados.

**MQRC\_INCOMPLETE\_GROUP**

(2241, X'8C1') Grupo de mensagens não concluído.

**MQRC\_INCOMPLETE\_MSG**

(2242, X'8C2') Mensagem lógica não concluída.

**MQRC\_PRIORITY\_EXCEEDS\_MAXIMUM**

(2049, X'801 ') A prioridade da mensagem excede o valor máximo suportado.

**MQRC\_UNKNOWN\_REPORT\_OPTION**

(2104, X'838 ') Opções de relatório no descritor de mensagens não reconhecidas.

Se CompCode for MQCC\_FAILED:

**MQRC\_ADAPTER\_NOT\_AVAILABLE**

(2204, X'89C') Adaptador não disponível.

**MQRC\_ADAPTER\_SERV\_LOAD\_ERROR**

(2130, X'852') Não foi possível carregar o módulo de serviço do adaptador.

**MQRC\_ALIAS\_BASE\_Q\_TYPE\_ERROR**

(2001, X'7D1') A fila base do alias não é um tipo válido.

**MQRC\_API\_EXIT\_ERROR**

(2374, X'946') Falha na saída de API.

**MQRC\_API\_EXIT\_LOAD\_ERROR**

(2183, X'887') Não foi possível carregar a saída de API.

**MQRC\_ASID\_MISMATCH**

(2157, X'86D') ASIDs de início e primárias são diferentes.

**MQRC\_BACKED\_OUT**

(2003, X'7D3') Unidade de trabalho retornada.

**MQRC\_BUFFER\_ERROR**

(2004, X'7D4') Parâmetro de buffer inválido.

**MQRC\_BUFFER\_LENGTH\_ERROR**

(2005, X'7D5') Parâmetro de comprimento de buffer inválido.

**MQRC\_CALL\_IN\_PROGRESS**

(2219, X'8AB') Chamada MQI inserida antes da chamada anterior concluída.

**MQRC\_CF\_NOT\_AVAILABLE**

(2345, X'929') recurso de acoplamento não disponível.

**MQRC\_CF\_STRUC\_AUTH\_FAILED**

(2348, X'92C') A verificação de autorização da estrutura do recurso de acoplamento falhou..

**MQRC\_CF\_STRUC\_ERROR**

(2349, X'92D') Estrutura de recurso de acoplamento não válida.

**MQRC\_CF\_STRUC\_FAILED**

(2373, X'945') Falha na estrutura do recurso de acoplamento.

**MQRC\_CF\_STRUC\_IN\_USE**

(2346, X'92A') Estrutura de recurso de acoplamento em uso.

**MQRC\_CF\_STRUC\_LIST\_HDR\_IN\_USE**

(2347, X'92B') Cabeçalho da lista da estrutura do recurso de acoplamento em uso.

**MQRC\_CFGR\_ERROR**

(2416, X'970') A estrutura do parâmetro do grupo PCF MQCFGR nos dados da mensagem não é válida..

**MQRC\_CFH\_ERROR**

(2235, X'8BB') Estrutura do cabeçalho PCF não válida.

**MQRC\_CFIF\_ERROR**

(2414, X'96E') A estrutura do parâmetro de filtro inteiro PCF nos dados da mensagem não é válida.

**MQRC\_CFIL\_ERROR**

(2236, X'8BC') A estrutura do parâmetro da lista de inteiros PCF ou a estrutura do parâmetro da lista de inteiros PCIF\*64 não é válida.

**MQRC\_CFIN\_ERROR**

(2237, X'8BD') Estrutura de parâmetro de número inteiro PCF ou estrutura de parâmetro de número inteiro PCIF\*64 não é válida.

**MQRC\_CFSF\_ERROR**

(2415, X'96F') A estrutura do parâmetro de filtro de sequência PCF nos dados da mensagem não é válida..

**MQRC\_CFSL\_ERROR**

(2238, X'8BE') Estrutura do parâmetro da lista de cadeias PCF não é válida.

**MQRC\_CFST\_ERROR**

(2239, X'8BF') Estrutura do parâmetro de sequência PCF não válida.

**MQRC\_CICS\_WAIT\_FAILED**

(2140, X'85C') Pedido de espera rejeitado por CICS.

**MQRC\_CLUSTER\_EXIT\_ERROR**

(2266, X'8DA') Falha na saída de carga de trabalho do cluster.

**MQRC\_CLUSTER\_RESOLUTION\_ERROR**

(2189, X'88D') Falha na resolução do nome do cluster.

**MQRC\_CLUSTER\_RESOURCE\_ERROR**

(2269, X'8DD') Erro de recurso de cluster

**MQRC\_COD\_NOT\_VALID\_FOR\_XCF\_Q**

(2106, X'83A') A opção de relatório COD não é válida para a fila XCF.

**MQRC\_CONNECTION\_BROKEN**

(2009, X'7D9') Conexão ao gerenciador de filas perdida.

**MQRC\_CONNECTION\_NOT\_AUTHORIZED**

(2217, X'8A9') Não autorizado para conexão.

**MQRC\_CONNECTION QUIESCING**

(2202, X'89A') Quiesce de conexão.

**MQRC\_CONNECTION\_STOPPING**

(2203, X'89B') Conexão sendo encerrada.

**MQRC\_CONTENT\_ERROR**

2554 (X'09FA') O conteúdo da mensagem não pôde ser analisado para determinar se a mensagem pode ser entregue a um assinante com um seletor de mensagem estendida

**MQRC\_CONTEXT\_HANDLE\_ERROR**

(2097, X'831 ') O identificador de fila referido não salva o contexto..

**MQRC\_CONTEXT\_NOT\_AVAILABLE**

(2098, X'832 ') Contexto não disponível para o identificador de fila referido.

**MQRC\_DATA\_LENGTH\_ERROR**

(2010, X'7DA') Parâmetro de comprimento de dados inválido.

**MQRC\_DB2\_NOT\_AVAILABLE**

(2342, X' 926 ') Db2 subsistema não disponível.

**MQRC\_DEF\_XMIT\_Q\_TYPE\_ERROR**

(2198, X'896 ') Fila de transmissão padrão não local.

**MQRC\_DEF\_XMIT\_Q\_USAGE\_ERROR**

(2199, X'897 ') Erro de uso da fila de transmissão padrão

**MQRC\_DH\_ERROR**

(2135, X'857 ') Estrutura do cabeçalho de distribuição inválida.

**MQRC\_DLH\_ERROR**

(2141, X'85D') Estrutura do cabeçalho de devoluções não válida.

**MQRC\_EPH\_ERROR**

(2420, X' 974 ') A estrutura PCF integrada não é válida.

**MQRC\_EXPIRY\_ERROR**

(2013, X'7DD') Tempo de expiração inválido.

**MQRC\_FEEDBACK\_ERROR**

(2014, X'7DE') Código de feedback inválido.

**MQRC\_GLOBAL\_UOW\_CONFLICT**

(2351, X'92F') Unidades de trabalho globais conflitam.

**MQRC\_GROUP\_ID\_ERROR**

(2258, X'8D2') Identificador de grupo inválido.

**MQRC\_HANDLE\_IN\_USE\_FOR\_UOW**

(2353, X'931') Manipulação em uso para unidade de trabalho global.

**MQRC\_HANDLE\_NOT\_AVAILABLE**

(2017, X'7E1') Não há mais identificadores disponíveis

**MQRC\_HCONN\_ERROR**

(2018, X'7E2') Manipulação de conexões não válida.

**MQRC\_HEADER\_ERROR**

(2142, X'85E') IBM MQ estrutura do cabeçalho não válida.

**MQRC\_IIH\_ERROR**

(2148, X'864 ') IMS estrutura do cabeçalho de informações não válida.

**MQRC\_LOCAL\_UOW\_CONFLICT**

(2352, X'930') Unidade de trabalho global conflita com unidade de trabalho local.

**MQRC\_MD\_ERROR**

(2026, X'7EA') Descritor de mensagem inválido.

**MQRC\_MDE\_ERROR**

(2248, X'8C8') Extensão do descritor de mensagens inválida.

**MQRC\_MISSING\_REPLY\_TO\_Q**

(2027, X'7EB') Fila de resposta ausente.

**MQRC\_MISSING\_WIH**

(2332, X'91C') Os dados da mensagem não começam com MQWIH.

**MQRC\_MSG\_FLAGS\_ERROR**

(2249, X'8C9') Sinalizadores de mensagem não válidos.

**MQRC\_MSG\_SEQ\_NUMBER\_ERROR**

(2250, X'8CA') Número de sequência da mensagem inválido.

**MQRC\_MSG\_TOO\_BIG\_FOR\_Q**

(2030, X'7EE') Comprimento da mensagem maior que o máximo para a fila.

**MQRC\_MSG\_TOO\_BIG\_FOR\_Q\_MGR**

(2031, X'7EF') Comprimento da mensagem maior que o máximo para o gerenciador de filas.

**MQRC\_MSG\_TYPE\_ERROR**

(2029, X'7ED') Tipo de mensagem no descritor de mensagens inválido.

**MQRC\_MULTIPLE\_REASONS**

(2136, X'858 ') Códigos de razão múltiplos retornados.

**MQRC\_NO\_DESTINATIONS\_AVAILABLE**

(2270, X'8DE') Nenhuma fila de destinos disponível.

**MQRC\_NOT\_AUTHORIZED**

(2035, X'7F3') Não autorizado para acesso.

**MQRC\_OBJECT\_DAMAGED**

(2101, X'835') Objeto danificado.

**MQRC\_OBJECT\_IN\_USE**

(2042, X'7FA') Objeto já aberto com opções conflitantes.

**MQRC\_OBJECT\_LEVEL\_INCOMPATÍVEL**

(2360, X' 938 ') Nível de objeto não compatível.

**MQRC\_OBJECT\_NAME\_ERROR**

(2152, X'868 ') Nome do objeto inválido.

**MQRC\_OBJECT\_NOT\_UNIQUE**

(2343, X' 927 ') Objeto não exclusivo.

**MQRC\_OBJECT\_Q\_MGR\_NAME\_ERROR**

(2153, X'869 ') O nome do gerenciador de filas de objeto não é válido.

**MQRC\_OBJECT\_RECORDS\_ERROR**

(2155, X'86B') Registros de objeto inválidos.

**MQRC\_OBJECT\_TYPE\_ERROR**

(2043, X'7FB') Tipo de objeto inválido.

**MQRC\_OD\_ERROR**

(2044, X'7FC') Estrutura do descritor de objeto inválida.

**MQRC\_OFFSET\_ERROR**

(2251, X'8CB') O deslocamento do segmento da mensagem não é válido..

**MQRC\_OPTIONS\_ERROR**

(2046, X'7FE') Opções não válidas ou não consistentes.

**MQRC\_ORIGINAL\_LENGTH\_ERROR**

(2252, X'8CC') Comprimento original inválido.

**MQRC\_PAGESET\_ERROR**

(2193, X'891') Erro ao acessar o conjunto de dados do conjunto de páginas.

**MQRC\_PAGESET\_FULL**

(2192, X'890') O meio de armazenamento externo está cheio.

**MQRC\_PCF\_ERROR**

(2149, X'865 ') estruturas PCF inválidas.

**MQRC\_PERSISTENCE\_ERROR**

(2047, X'7FF') Persistência inválida.



**MQRC\_PERSISTENT\_NOT\_ALLOWED**

(2048, X'800 ') A Fila não suporta mensagens persistentes.

**MQRC\_PMO\_ERROR**

(2173, X'87D') Estrutura de opções de mensagem de entrada inválida.

**MQRC\_PMO\_RECORD\_FLAGS\_ERROR**

(2158, X'86E') Sinalizadores de registro de mensagem de colocação não válidos.

**MQRC\_PRIORITY\_ERROR**

(2050, X'802 ') A prioridade da mensagem não é válida.

**MQRC\_PUBLICATION\_FAILURE**

(2502, X'9C6') A publicação não foi entregue a nenhum dos assinantes.

**MQRC\_PUT\_INHIBITED**

(2051, X'803 ') Chamadas de put inibidas para a fila

**MQRC\_PUT\_MSG\_RECORDS\_ERROR**

(2159, X'86F') Os registros de mensagens Put não são válidos.

**MQRC\_Q\_DELETED**

(2052, X'804') A fila foi excluída.

**MQRC\_Q\_FULL**

(2053, X'805 ') A fila já contém número máximo de mensagens.

**MQRC\_Q\_MGR\_NAME\_ERROR**

(2058, X'80A') Nome do gerenciador de filas não válido ou não conhecido.

**MQRC\_Q\_MGR\_NOT\_AVAILABLE**

(2059, X'80B') Gerenciador de filas não disponível para conexão.

**MQRC\_Q\_MGR QUIESCING**

(2161, X'871') Quiesce do gerenciador de filas.

**MQRC\_Q\_MGR\_STOPPING**

(2162, X'872') Gerenciador de filas sendo encerrado.

**MQRC\_Q\_SPACE\_NOT\_AVAILABLE**

(2056, X'808 ') Nenhum espaço disponível no disco para fila.

**MQRC\_Q\_TYPE\_ERROR**

(2057, X'809 ') Tipo de fila inválido.

**MQRC\_RECS\_PRESENT\_ERROR**

(2154, X'86A') Número de registros presentes não válidos.

**MQRC\_REMOTE\_Q\_NAME\_ERROR**

(2184, X'888 ') Nome da fila remota inválido.

**MQRC\_REPORT\_OPTIONS\_ERROR**

(2061, X'80D') Opções de relatório no descritor de mensagem não válidas.

**MQRC\_RESOURCE\_PROBLEM**

(2102, X'836') Recursos insuficientes do sistema disponíveis.

**MQRC\_RESPONSE\_RECORDS\_ERROR**

(2156, X'86C') Registros de resposta inválidos.

**MQRC\_RFH\_ERROR**

(2334, X'91E') A estrutura MQRFH ou MQRFH2 não é válida.

**MQRC\_RMH\_ERROR**

(2220, X'8AC') A estrutura do cabeçalho da mensagem de referência não é válida.

**MQRC\_SECURITY\_ERROR**

(2063, X'80F') Ocorreu um erro na segurança.

**MQRC\_SEGMENT\_LENGTH\_ZERO**

(2253, X'8CD') O comprimento dos dados no segmento de mensagem é zero...

**MQRC\_SELECTION\_NOT\_AVAILABLE**

2551 (X'09F7') Existe um possível assinante para a publicação, mas o gerenciador de filas não pode verificar se a publicação deve ser enviada para o assinante.

**MQRC\_STOPPED\_BY\_CLUSTER\_EXIT**

(2188, X'88C') Chamada rejeitada pela saída de carga de trabalho do cluster.

**MQRC\_STORAGE\_CLASS\_ERROR**

(2105, X'839 ') Erro da classe de armazenamento..

**MQRC\_STORAGE\_MEDIUM\_FULL**

(2192, X'890') O meio de armazenamento externo está cheio.

**MQRC\_STORAGE\_NOT\_AVAILABLE**

(2071, X'817') Armazenamento insuficiente disponível.

**MQRC\_SUPPRESSED\_BY\_EXIT**

(2109, X'83D') Chamada suprimida pelo programa de saída.

**MQRC\_SYNCPOINT\_LIMIT\_REACHED**

(2024, X'7E8') Nenhuma mensagem adicional pode ser manipulada dentro da unidade de trabalho atual.

**MQRC\_SYNCPOINT\_NOT\_AVAILABLE**

(2072, X'818 ') Suporte de ponto de sincronização não disponível.

**MQRC\_TM\_ERROR**

(2265, X'8D9') Estrutura da mensagem do acionador inválida.

**MQRC\_TMC\_ERROR**

(2191, X'88F') A estrutura da mensagem do acionador de caracteres não é válida..

**MQRC\_UNEXPECTED\_ERROR**

(2195, X'893') Ocorreu um erro inesperado.

**MQRC\_UNKNOWN\_ALIAS\_BASE\_Q**

(2082, X'822 ') Fila base de alias desconhecida.

**MQRC\_UNKNOWN\_DEF\_XMIT\_Q**

(2197, X'895 ') Fila de transmissão padrão desconhecida.

**MQRC\_UNKNOWN\_OBJECT\_NAME**

(2085, X'825 ') Nome do objeto desconhecido.

**MQRC\_UNKNOWN\_OBJECT\_Q\_MGR**

(2086, X'826 ') Gerenciador de filas de objeto desconhecido.

**MQRC\_UNKNOWN\_REMOTE\_Q\_MGR**

(2087, X'827 ') Gerenciador de filas remotas desconhecido.

**MQRC\_UNKNOWN\_XMIT\_Q**

(2196, X'894 ') Fila de transmissão desconhecida.

**MQRC\_UOW\_ENLISTMENT\_ERROR**

(2354, X'932') Cadastramento na unidade de trabalho global falhou.

**MQRC\_UOW\_MIX\_NOT\_SUPPORTED**

(2355, X'933') Combinação de chamadas de unidade de trabalho não suportada.

**MQRC\_UOW\_NOT\_AVAILABLE**

(2255, X'8CF') Unidade de trabalho não disponível para o gerenciador de filas a usar.

**MQRC\_WIH\_ERROR**

(2333, X'91D') Estrutura MQWIH não válida.

**MQRC\_ERRR\_CF\_LEVEL**

(2366, X'93E') A estrutura do recurso de acoplamento está no nível errado.

**MQRC\_WRONG\_MD\_VERSION**

(2257, X'8D1') Versão errada do MQMD fornecido.

**MQRC\_XMIT\_Q\_TYPE\_ERROR**

(2091, X'82B') Fila de transmissão não local.

## **MQRC\_XMIT\_Q\_USAGE\_ERROR**

(2092, X'82C') Fila de transmissão com uso errado.

## **MQRC\_XQH\_ERROR**

(2260, X'8D4') A estrutura do cabeçalho da fila de transmissão não é válida.

Para obter informações detalhadas sobre esses códigos, consulte [Mensagens e códigos de razão](#).

## **Observações de Uso**

1. As chamadas MQPUT e MQPUT1 podem ser usadas para colocar mensagens em uma fila; a chamada a ser usada depende das circunstâncias:
  - Use a chamada MQPUT para colocar várias mensagens na *mesma fila*  
Uma chamada MQOPEN especificando a opção MQOO\_OUTPUT é emitida primeiro, seguida por uma ou mais solicitações MQPUT para incluir mensagens na fila; finalmente, a fila é fechada com uma chamada MQCLOSE. Isso fornece melhor desempenho do que o uso repetido da chamada MQPUT1 ..
  - Use a chamada MQPUT1 para colocar apenas *uma* mensagem em uma fila.  
Essa chamada encapsula as chamadas MQOPEN, MQPUT e MQCLOSE em uma única chamada, minimizando o número de chamadas que devem ser emitidas..
2. Se um aplicativo colocar uma sequência de mensagens na mesma fila sem usar grupos de mensagens, a ordem dessas mensagens será preservada desde que determinadas condições sejam satisfeitas. No entanto, na maioria dos ambientes, a chamada MQPUT1 não satisfaz essas condições e, portanto, não preserva a ordem de mensagens. A chamada MQPUT deve ser usada nesses ambientes. Consulte [Observações de Uso de MQPUT](#) para obter detalhes.
3. A chamada MQPUT1 pode ser usada para colocar mensagens em listas de distribuição. Para obter informações gerais sobre isso, consulte as notas de uso das chamadas MQOPEN e MQPUT.

As listas de distribuição são suportadas nos seguintes ambientes:

-  AIX
-  IBM i
-  Linux
-  Windows

e para clientes IBM MQ conectados a esses sistemas.

As diferenças a seguir se aplicam ao usar a chamada MQPUT1 :

- a. Se o aplicativo fornecer registros de resposta MQRR, eles deverão ser fornecidos usando a estrutura MQOD; eles não poderão ser fornecidos usando a estrutura MQPMO
- b. O código de razão MQRC\_OPEN\_FAILED nunca é retornado pelo MQPUT1 nos registros de resposta; se uma fila falhar ao abrir, o registro de resposta para essa fila contém o código de razão resultante da operação de abertura.

Se uma operação aberta para uma fila for bem-sucedida com um código de conclusão de MQCC\_WARNING, o código de conclusão e o código de razão no registro de resposta para essa fila serão substituídos pela conclusão e os códigos de razão resultantes da operação put.

Como com as chamadas MQOPEN e MQPUT, o gerenciador de filas configura os registros de resposta (se fornecidos) somente quando o resultado da chamada não é o mesmo para todas as filas na lista de distribuição; isso é indicado pela chamada que conclui com o código de razão MQRC\_MULTIPLE\_REASON..

4. Se a chamada MQPUT1 for usada para colocar uma mensagem em uma fila de clusters, a chamada se comporta como se MQOO\_BIND\_NOT\_FIXED tivesse sido especificado na chamada MQOPEN.
5. Se uma mensagem for colocada com uma ou mais estruturas de cabeçalho do IBM MQ no início dos dados da mensagem do aplicativo, o gerenciador de filas executará determinadas verificações nas

estruturas de cabeçalho para verificar se elas são válidas Para obter mais informações sobre isso, consulte as notas de uso para a chamada MQPUT

6. Se mais de uma das situações de aviso surgir (consulte o parâmetro **CompCode** ), o código de razão retornado será o primeiro na lista a seguir que se aplica:

- a. MQRC\_MULTIPLE\_REASONS
- b. MQRC\_INCOMPLETE\_MSG
- c. MQRC\_INCOMPLETE\_GROUP
- d. MQRC\_PRIORITY\_EXCEEDS\_MAXIMUM ou MQRC\_UNKNOWN\_REPORT\_OPTION

7. Para a linguagem de programação Visual Basic, os seguintes pontos se aplicam:

- Se o tamanho do parâmetro **Buffer** for menor que o comprimento especificado pelo parâmetro **BufferLength** , a chamada falhará com código de razão MQRC\_BUFFER\_LENGTH\_ERROR.
- O parâmetro **Buffer** é declarado como sendo do tipo String Se os dados a serem colocados na fila não forem do tipo String, use oMQPUT1Any no lugar de MQPUT1.

A chamada MQPUT1Any tem os mesmos parâmetros da chamada MQPUT1 , exceto que o parâmetro **Buffer** é declarado como sendo do tipo Any, permitindo que qualquer tipo de dados seja colocado na fila. No entanto, isso significa que Buffer não pode ser verificado para assegurar que tenha pelo menos BufferLength bytes de tamanho.

8. Quando uma chamada MQPUT1 for emitida com MQPMO\_SYNCPOINT, o comportamento padrão muda, de forma que a operação put seja concluída de forma assíncrona. Isso pode causar uma mudança no comportamento de alguns aplicativos que dependem de determinados campos nas estruturas MQOD e MQMD que estão sendo retornadas, mas que agora contêm valores não definidos. Um aplicativo pode especificar MQPMO\_SYNC\_RESPONSE para assegurar que a operação de colocação seja executada de forma síncrona e que todos os valores de campo apropriados sejam concluídos

## Chamada C

```
MQPUT1 (Hconn, &ObjDesc, &MsgDesc, &PutMsgOpts,  
        BufferLength, Buffer, &CompCode, &Reason);
```

Declare os parâmetros conforme a seguir:

```
MQHCONN  Hconn;           /* Connection handle */  
MQOD     ObjDesc;        /* Object descriptor */  
MQMD     MsgDesc;        /* Message descriptor */  
MQPMO    PutMsgOpts;     /* Options that control the action of MQPUT1 */  
MQLONG   BufferLength;    /* Length of the message in Buffer */  
MQBYTE   Buffer[n];       /* Message data */  
MQLONG   CompCode;       /* Completion code */  
MQLONG   Reason;         /* Reason code qualifying CompCode */
```

## Chamada COBOL

```
CALL 'MQPUT1' USING HCONN, OBJDESC, MSGDESC, PUTMSGOPTS,  
                   BUFFERLENGTH, BUFFER, COMPCODE, REASON.
```

Declare os parâmetros conforme a seguir:

```
** Connection handle  
01 HCONN          PIC S9(9) BINARY.  
** Object descriptor  
01 OBJDESC.  
   COPY CMQODV.  
** Message descriptor  
01 MSGDESC.  
   COPY CMQMDV.
```

```

** Options that control the action of MQPUT1
01 PUTMSGOPTS.
   COPY CMQPMOV.
** Length of the message in BUFFER
01 BUFFERLENGTH PIC S9(9) BINARY.
** Message data
01 BUFFER PIC X(n).
** Completion code
01 COMPCODE PIC S9(9) BINARY.
** Reason code qualifying COMPCODE
01 REASON PIC S9(9) BINARY.

```

## Chamada PL/I

```

call MQPUT1 (Hconn, ObjDesc, MsgDesc, PutMsgOpts, BufferLength, Buffer,
            CompCode, Reason);

```

Declare os parâmetros conforme a seguir:

```

dcl Hconn          fixed bin(31); /* Connection handle */
dcl ObjDesc        like MQOD;    /* Object descriptor */
dcl MsgDesc        like MQMD;    /* Message descriptor */
dcl PutMsgOpts     like MQPMO;   /* Options that control the action of
                                MQPUT1 */
dcl BufferLength    fixed bin(31); /* Length of the message in Buffer */
dcl Buffer          char(n);      /* Message data */
dcl CompCode       fixed bin(31); /* Completion code */
dcl Reason         fixed bin(31); /* Reason code qualifying CompCode */

```

## Chamada do Assembler de Alto Nível

```

CALL MQPUT1, (HCONN,OBJDESC,MSGDESC,PUTMSGOPTS,BUFFERLENGTH, X
            BUFFER,COMPCODE,REASON)

```

Declare os parâmetros conforme a seguir:

```

HCONN          DS      F      Connection handle
OBJDESC        CMQODA   ,      Object descriptor
MSGDESC        CMQMDA   ,      Message descriptor
PUTMSGOPTS     CMQPMOA  ,      Options that control the action of MQPUT1
BUFFERLENGTH   DS      F      Length of the message in BUFFER
BUFFER         DS      CL(n)  Message data
COMPCODE       DS      F      Completion code
REASON         DS      F      Reason code qualifying COMPCODE

```

## Chamada do Visual Basic

Windows

```

MQPUT1 Hconn, ObjDesc, MsgDesc, PutMsgOpts, BufferLength, Buffer,
      CompCode, Reason

```

Declare os parâmetros conforme a seguir:

```

Dim Hconn          As Long   'Connection handle'
Dim ObjDesc        As MQOD   'Object descriptor'
Dim MsgDesc        As MQMD   'Message descriptor'
Dim PutMsgOpts     As MQPMO  'Options that control the action of MQPUT1'
Dim BufferLength    As Long   'Length of the message in Buffer'
Dim Buffer          As String 'Message data'
Dim CompCode       As Long   'Completion code'
Dim Reason         As Long   'Reason code qualifying CompCode'

```

## MQSET - configurar atributos do objeto

Use a chamada MQSET para alterar os atributos de um objeto representado por uma manipulação O objeto deve ser uma fila..

### Sintaxe


MQSET (*Hconn*, *Hobj*, *SelectorCount*, *Seletores*, *IntAttrCount*, *IntAttrs*, *CharAttrLength*, *CharAttrs*, *Compcode*, *Motivo*)

### Parâmetros

#### Hconn

Tipo: MQHCONN - entrada

Esta manipulação representa a conexão ao gerenciador de filas. O valor de Hconn foi retornado por uma chamada anterior MQCONN ou MQCONNX.

 Em z/OS para aplicativos CICS, a chamada MQCONN pode ser omitida e o seguinte valor especificado para Hconn:

#### MQHC\_DEF\_HCONN

Manipulação de conexões padrão.

#### Hobj

Tipo: MQHOBJ - entrada

Essa manipulação representa o objeto da fila com atributos que devem ser configurados O identificador foi retornado por uma chamada MQOPEN anterior que especificou a opção MQOO\_SET..

#### SelectorCount

Tipo: MQLONG - entrada

Essa é a contagem de seletores fornecidos na matriz *Selectors*. É o número de atributos a serem configurados. Zero é um valor válido. O número máximo permitido é 256..

#### Seletores

Tipo: MQLONGxSelectorContagem-entrada

Esta é uma matriz de seletores de atributo **SelectorCount**; cada seletor identifica um atributo (número inteiro ou caractere) com um valor que deve ser configurado.

Cada seletor deve ser válido pelo tipo de fila que *Hobj* representa. Apenas determinados valores MQIA\_\* e MQCA\_\* são permitidos; conforme listado posteriormente.

Os seletores podem ser especificados em qualquer ordem Os valores de atributo que correspondem a seletores de atributo de número inteiro (seletores MQIA\_\*) devem ser especificados em *IntAttrs* na mesma ordem em que esses seletores ocorrem em *Selectors*. Os valores de atributo que correspondem aos seletores de atributo de caractere (seletores MQCA\_\*) devem ser especificados em *CharAttrs* na mesma ordem em que esses seletores ocorrem.. Os seletores MQIA\_\* podem ser intercalados com os seletores MQCA\_\*; somente a ordem relativa dentro de cada tipo é importante.

É possível especificar o mesmo seletor mais de uma vez; se o fizer, o último valor especificado para um seletor específico será aquele que entra em vigor.

#### Nota:

1. Os seletores de atributo de número inteiro e caractere são alocados em dois intervalos diferentes; os seletores MQIA\_\* residem no intervalo MQIA\_FIRST a MQIA\_LAST e os seletores MQCA\_\* no intervalo MQCA\_FIRST a MQCA\_LAST.

Para cada intervalo, as constantes MQIA\_LAST\_USED e MQCA\_LAST\_USED definem o valor mais alto que o gerenciamento de filas aceita.

2. Se todos os seletores MQIA\_\* ocorrerem primeiro, os mesmos números de elementos poderão ser usados para abordar elementos correspondentes nas matrizes *Selectors* e *IntAttrs*.

3. Se o parâmetro **SelectorCount** for zero, **Selectors** não será referido; nesse caso, o endereço do parâmetro transmitido por programas gravados no assembler C ou System/390 poderá ser nulo.

Os atributos que podem ser configurados são listados na tabela a seguir: Nenhum outro atributo pode ser configurado utilizando esta chamada Para os seletores de atributo MQCA\_\*, a constante que define o comprimento em bytes da cadeia necessária em CharAttrr é fornecida entre parênteses.

*Tabela 554. seletores de atributos MQSET para filas*

Seletor	Descrição	Nota
MQCA_TRIGGER_DATA	Dados do acionador (MQ_TRIGGER_DATA_LENGTH).	
MQIA_DIST_LISTS	Suporte à lista de distribuições	1
MQIA_INIBBIT_GET	Se operações get são permitidas.	
MQIA_INIBBIT_PUT	Se operações put são permitidas.	
MQIA_TRIGGER_CONTROL	Acionador de controle.	
MQIA_TRIGGER_DEPTH	Profundidade do acionador.	
MQIA_TRIGGER_MSG_PRIORITY	Prioridade da mensagem limite para acionadores.	
MQIA_TRIGGER_TYPE	Tipo de acionador.	

**Nota:**

1. Suportado apenas nas seguintes plataformas:

-  AIX
-  IBM i
-  Linux
-  Windows

e para o IBM MQ MQI clients conectado a esses sistemas

**IntAttrCount**

Tipo: MQLONG - entrada

Este é o número de elementos na matriz IntAttrr e deve ser pelo menos o número de seletores MQIA\_\* no parâmetro **Selectors**. Zero é um valor válido se não houver nenhum.

**IntAttrr**

Tipo: MQLONGxIntAttrCount - entrada

Essa é uma matriz de valores de atributo de número inteiro IntAttrCount. Esses valores de atributo devem estar na mesma ordem que os seletores MQIA\_\* na matriz Selectors

Se o parâmetro **IntAttrCount** ou **SelectorCount** for zero, IntAttrr não será referido; nesse caso, o endereço do parâmetro transmitido por programas gravados no assembler C ou System/390 poderá ser nulo.

**Comprimento de CharAttr**

Tipo: MQLONG - entrada

Esse é o comprimento em bytes do parâmetro CharAttrr e deve ser pelo menos a soma dos comprimento dos atributos de caractere especificados na matriz Selectors. Zero será um valor válido se não houver seletores MQCA\_\* em Selectors.

**CharAttrr**

Tipo: MQCHAR x CharAttrComprimento-entrada

Este é o buffer que contém os valores de atributo de caractere, concatenados juntos. O comprimento do buffer é fornecido pelo parâmetro **CharAttrLength**.

Os atributos de caracteres devem ser especificados na mesma ordem que os seletores MQCA\_\* na matriz **Selectors**. O comprimento de cada atributo de caractere é fixo (consulte **Seletores**). Se o valor a ser configurado para um atributo contiver menos caracteres não em branco do que o comprimento definido do atributo, preencha o valor em **CharAttrs** à direita com espaços em branco para que o valor do atributo corresponda ao comprimento definido do atributo.

Se o parâmetro **CharAttrLength** ou **SelectorCount** for zero, **CharAttrs** não será referido; nesse caso, o endereço do parâmetro transmitido por programas gravados no assembler C ou System/390 poderá ser nulo.

### **CompCode**

Tipo: MQLONG - saída

O código de conclusão; é um dos seguintes:

#### **MQCC\_OK**

Indica conclusão bem-sucedida.

#### **MQCC\_FAILED**

A chamada falhou.

### **Razão**

Tipo: MQLONG - saída

O código de razão qualificando **CompCode**.

Se **CompCode** for **MQCC\_OK**:

#### **MQRC\_NONE**

(0, X'000') Nenhuma razão a relatar.

Se **CompCode** for **MQCC\_FAILED**:

#### **MQRC\_ADAPTER\_NOT\_AVAILABLE**

(2204, X'89C') Adaptador não disponível.

#### **MQRC\_ADAPTER\_SERV\_LOAD\_ERROR**

(2130, X'852') Não foi possível carregar o módulo de serviço do adaptador.

#### **MQRC\_API\_EXIT\_ERROR**

(2374, X'946') Falha na saída de API.

#### **MQRC\_API\_EXIT\_LOAD\_ERROR**

(2183, X'887') Não foi possível carregar a saída de API.

#### **MQRC\_ASID\_MISMATCH**

(2157, X'86D') ASIDs de início e primárias são diferentes.

#### **MQRC\_CALL\_IN\_PROGRESS**

(2219, X'8AB') Chamada MQI inserida antes da chamada anterior concluída.

#### **MQRC\_CF\_NOT\_AVAILABLE**

(2345, X'929') Recurso de acoplamento não disponível.

#### **MQRC\_CF\_STRUC\_FAILED**

(2373, X'945') Falha na estrutura do recurso de acoplamento.

#### **MQRC\_CF\_STRUC\_IN\_USE**

(2346, X'92A') Estrutura de recurso de acoplamento em uso.

#### **MQRC\_CF\_STRUC\_LIST\_HDR\_IN\_USE**

(2347, X'92B') Cabeçalho da lista da estrutura do recurso de acoplamento em uso.

#### **MQRC\_CHAR\_ATTR\_LENGTH\_ERROR**

(2006, X'7D6') Comprimento de atributos de caractere não válido.

#### **MQRC\_CHAR\_ATTRS\_ERROR**

(2007, X'7D7') Sequência de atributos de caractere não válida.



**MQRC\_CICS\_WAIT\_FAILED**  
(2140, X'85C') Pedido de espera rejeitado por CICS.

**MQRC\_CONNECTION\_BROKEN**  
(2009, X'7D9') Conexão ao gerenciador de filas perdida.

**MQRC\_CONNECTION\_NOT\_AUTHORIZED**  
(2217, X'8A9') Não autorizado para conexão.

**MQRC\_CONNECTION\_STOPPING**  
(2203, X'89B') Conexão sendo encerrada.

**MQRC\_DB2\_NOT\_AVAILABLE**  
(2342, X' 926 ') Db2 subsistema não disponível.

**MQRC\_HCONN\_ERROR**  
(2018, X'7E2') Manipulação de conexões não válida.

**MQRC\_HOBJ\_ERROR**  
(2019, X'7E3') Manipulação de objetos não válida.

**MQRC\_INHIBIT\_VALUE\_ERROR**  
(2020, X'7E4') O valor para o atributo da fila inibir-obter ou inibir-colocar não é válido.

**MQRC\_INT\_ATTR\_COUNT\_ERROR**  
(2021, X'7E5') Contagem de atributos de número inteiro inválida.

**MQRC\_INT\_ATTRS\_ARRAY\_ERROR**  
(2023, X'7E7') A matriz de atributos de número inteiro não é válida.

**MQRC\_NOT\_OPEN\_FOR\_SET**  
(2040, X'7F8') Fila não aberta para configurar.

**MQRC\_OBJECT\_CHANGED**  
(2041, X'7F9') Definição de objeto alterada desde a abertura.

**MQRC\_OBJECT\_DAMAGED**  
(2101, X'835') Objeto danificado.

**MQRC\_PAGESET\_ERROR**  
(2193, X'891') Erro ao acessar o conjunto de dados do conjunto de páginas.

**MQRC\_Q\_DELETED**  
(2052, X'804') A fila foi excluída.

**MQRC\_Q\_MGR\_NAME\_ERROR**  
(2058, X'80A') Nome do gerenciador de filas não válido ou não conhecido.

**MQRC\_Q\_MGR\_NOT\_AVAILABLE**  
(2059, X'80B') Gerenciador de filas não disponível para conexão.

**MQRC\_Q\_MGR\_STOPPING**  
(2162, X'872') Gerenciador de filas sendo encerrado.

**MQRC\_RESOURCE\_PROBLEM**  
(2102, X'836') Recursos insuficientes do sistema disponíveis.

**MQRC\_SELECTOR\_COUNT\_ERROR**  
(2065, X'811 ') Contagem de seletores inválida.

**MQRC\_SELECTOR\_ERROR**  
(2067, X'813 ') Seletor de atributo inválido.

**MQRC\_SELECTOR\_LIMIT\_EXCEDIDO**  
(2066, X'812 ') Contagem de seletores muito grande.

**MQRC\_STORAGE\_NOT\_AVAILABLE**  
(2071, X'817') Armazenamento insuficiente disponível.

**MQRC\_SUPPRESSED\_BY\_EXIT**  
(2109, X'83D') Chamada suprimida pelo programa de saída.

**MQRC\_TRIGGER\_CONTROL\_ERROR**  
(2075, X'81B') O valor para o atributo trigger-control não é válido.

**MQRC\_TRIGGER\_DEPTH\_ERROR**

(2076, X'81C') Valor para atributo de profundidade do acionador não válido.

**MQRC\_TRIGGER\_MSG\_PRIORITY\_ERR**

(2077, X'81D') Valor para o atributo trigger-message-priority não válido.

**MQRC\_TRIGGER\_TYPE\_ERROR**

(2078, X'81E') O valor para o atributo trigger-type não é válido.

**MQRC\_UNEXPECTED\_ERROR**

(2195, X'893') Ocorreu um erro inesperado.

Para obter informações detalhadas sobre esses códigos, consulte [Mensagens e códigos de razão](#).

## Observações de Uso

1. Usando esta chamada, o aplicativo pode especificar uma matriz de atributos de número inteiro ou uma coleção de sequências de atributos de caracteres ou ambos. Se não ocorrerem erros, os atributos especificados serão todos configurados simultaneamente. Se ocorrer um erro (por exemplo, se um seletor não for válido ou for feita uma tentativa de configurar um atributo para um valor inválido), a chamada falhará e nenhum atributo será configurado.

2. Os valores de atributos podem ser determinados usando a chamada MQINQ; consulte [“MQINQ- Consultar atributos do objeto.”](#) na página 730 para obter detalhes

**Nota:** Nem todos os atributos com valores que podem ser consultados usando a chamada MQINQ podem ter seus valores alterados usando a chamada MQSET. Por exemplo, nenhum atributo do objeto de processo ou do gerenciador de fila pode ser configurado com essa chamada

3. As mudanças de atributo são preservadas em reinicializações do gerenciador de filas (além de alterações em filas dinâmicas temporárias, que não sobrevivem às reinicializações do gerenciador de filas)..
4. Não é possível alterar os atributos de uma fila de modelo usando a chamada MQSET. No entanto, se você abrir uma fila modelo usando a chamada MQOPEN com a opção MQOO\_SET, será possível usar a chamada MQSET para configurar os atributos da fila local dinâmica criada pela chamada MQOPEN.
5. Se o objeto que está sendo configurado for uma fila de clusters, deve haver uma instância local da fila de clusters para que a abertura seja bem-sucedida

Para obter mais informações sobre atributos de objeto, consulte:

- [“Atributos para filas”](#) na página 865
- [“Atributos para Listas de Nomes”](#) na página 899
- [“Atributos para Definições de Processo”](#) na página 901
- [“Atributos do gerenciador de filas”](#) na página 827

## Chamada C

```
MQSET (Hconn, Hobj, SelectorCount, Selectors, IntAttrCount, IntAttrs,
CharAttrLength, CharAttrs, &CompCode, &Reason);
```

Declare os parâmetros conforme a seguir:

```
MQHCONN  Hconn;           /* Connection handle */
MQHOBJ   Hobj;           /* Object handle */
MQLONG   SelectorCount;  /* Count of selectors */
MQLONG   Selectors[n];   /* Array of attribute selectors */
MQLONG   IntAttrCount;   /* Count of integer attributes */
MQLONG   IntAttrs[n];    /* Array of integer attributes */
MQLONG   CharAttrLength; /* Length of character attributes buffer */
MQCHAR   CharAttrs[n];   /* Character attributes */
MQLONG   CompCode;       /* Completion code */
MQLONG   Reason;         /* Reason code qualifying CompCode */
```

## Chamada COBOL

```
CALL 'MQSET' USING HCONN, HOBJ, SELECTORCOUNT, SELECTORS-TABLE,  
                  INTATTRCOUNT, INTATTRS-TABLE, CHARATTRLENGTH,  
                  CHARATTRS, COMPCODE, REASON.
```

Declare os parâmetros conforme a seguir:

```
** Connection handle  
01 HCONN          PIC S9(9) BINARY.  
** Object handle  
01 HOBJ          PIC S9(9) BINARY.  
** Count of selectors  
01 SELECTORCOUNT PIC S9(9) BINARY.  
** Array of attribute selectors  
01 SELECTORS-TABLE.  
   02 SELECTORS   PIC S9(9) BINARY OCCURS n TIMES.  
** Count of integer attributes  
01 INTATTRCOUNT PIC S9(9) BINARY.  
** Array of integer attributes  
01 INTATTRS-TABLE.  
   02 INTATTRS   PIC S9(9) BINARY OCCURS n TIMES.  
** Length of character attributes buffer  
01 CHARATTRLENGTH PIC S9(9) BINARY.  
** Character attributes  
01 CHARATTRS      PIC X(n).  
** Completion code  
01 COMPCODE       PIC S9(9) BINARY.  
** Reason code qualifying COMPCODE  
01 REASON         PIC S9(9) BINARY.
```

## Chamada PL/I

```
call MQSET (Hconn, Hobj, SelectorCount, Selectors, IntAttrCount,  
            IntAttrs, CharAttrLength, CharAttrs, CompCode, Reason);
```

Declare os parâmetros conforme a seguir:

```
dcl Hconn          fixed bin(31); /* Connection handle */  
dcl Hobj          fixed bin(31); /* Object handle */  
dcl SelectorCount  fixed bin(31); /* Count of selectors */  
dcl Selectors(n)  fixed bin(31); /* Array of attribute selectors */  
dcl IntAttrCount  fixed bin(31); /* Count of integer attributes */  
dcl IntAttrs(n)   fixed bin(31); /* Array of integer attributes */  
dcl CharAttrLength fixed bin(31); /* Length of character attributes  
                                buffer */  
dcl CharAttrs     char(n); /* Character attributes */  
dcl CompCode      fixed bin(31); /* Completion code */  
dcl Reason        fixed bin(31); /* Reason code qualifying  
                                CompCode */
```

## Chamada do Assembler de Alto Nível

```
CALL MQSET, (HCONN, HOBJ, SELECTORCOUNT, SELECTORS, INTATTRCOUNT, X  
            INTATTRS, CHARATTRLENGTH, CHARATTRS, COMPCODE, REASON)
```

Declare os parâmetros conforme a seguir:

HCONN	DS	F	Connection handle
HOBJ	DS	F	Object handle
SELECTORCOUNT	DS	F	Count of selectors
SELECTORS	DS	(n)F	Array of attribute selectors
INTATTRCOUNT	DS	F	Count of integer attributes
INTATTRS	DS	(n)F	Array of integer attributes
CHARATTRLENGTH	DS	F	Length of character attributes buffer
CHARATTRS	DS	CL(n)	Character attributes

COMPCODE	DS	F	Completion code
REASON	DS	F	Reason code qualifying COMPCODE

## Chamada do Visual Basic

```
MQSET Hconn, Hobj, SelectorCount, Selectors, IntAttrCount, IntAttrs,
CharAttrLength, CharAttrs, CompCode, Reason
```

Declare os parâmetros conforme a seguir:

```
Dim Hconn As Long 'Connection handle'
Dim Hobj As Long 'Object handle'
Dim SelectorCount As Long 'Count of selectors'
Dim Selectors As Long 'Array of attribute selectors'
Dim IntAttrCount As Long 'Count of integer attributes'
Dim IntAttrs As Long 'Array of integer attributes'
Dim CharAttrLength As Long 'Length of character attributes buffer'
Dim CharAttrs As String 'Character attributes'
Dim CompCode As Long 'Completion code'
Dim Reason As Long 'Reason code qualifying CompCode'
```

## MQSETMP-Configurar propriedade de mensagem

Use a chamada MQSETMP para definir ou modificar uma propriedade de um identificador de mensagem

### Sintaxe

MQSETMP (*Hconn, Hmsg, SetProp, Nome, PropDesc, Tipo, ValueLength, Value, Compcode, Reason*)

### Parâmetros

#### Hconn

Tipo: MQHCONN - entrada

Esta manipulação representa a conexão ao gerenciador de filas.

O valor deve corresponder ao identificador de conexão que foi usado para criar o identificador de mensagem especificado no parâmetro **Hmsg** ... Se a manipulação de mensagem foi criada usando MQHC\_UNASSOCIATED\_HCONN, uma conexão válida deverá ser estabelecida na configuração de encadeamento de uma propriedade da manipulação de mensagens, caso contrário, a chamada falhará com o código de razão MQRC\_CONNECTION\_BROKEN

#### Hmsg

Tipo: MQHMSG-entrada

Esta é a manipulação de mensagem a ser modificada O valor foi retornado por uma chamada MQCRTMH anterior.

#### SetPropOpts

Tipo: MQSMPO-entrada

Controle como as propriedades de mensagem são configuradas

Essa estrutura permite que os aplicativos especifiquem opções que controlam como as propriedades de mensagem são configuradas A estrutura é um parâmetro de entrada na chamada MQSETMP..

Consulte [MQSMPO](#) para obter informações adicionais

#### Nome

Tipo: MQCHARV-entrada

Este é o nome da propriedade a ser configurada

Consulte [Nomes de Propriedade e Restrições de Nome de Propriedade](#) para obter informações adicionais sobre o uso de nomes de propriedade

## PropDesc

Tipo: MQPD-entrada / saída

Esta estrutura é utilizada para definir os atributos de uma propriedade, incluindo:

- o que acontece se a propriedade não for suportada
- a qual contexto de mensagem a propriedade pertence
- para quais mensagens a propriedade é copiada à medida que flui

Consulte [MQPD](#) para obter informações adicionais sobre esta estrutura..

## tipo

Tipo: MQLONG - entrada

O tipo de dados da propriedade sendo configurada. Pode ser um dos seguintes:

### **MQTYPE\_BOOLEAN**

Um booleano. *ValueLength* deve ser 4.

### **MQTYPE\_BYTE\_STRING**

Uma sequência de bytes. *ValueLength* deve ser zero ou maior.

### **MQTYPE\_INT8**

Um número inteiro assinado de 8 bits. *ValueLength* deve ser 1.

### **MQTYPE\_INT16**

Um número inteiro assinado de 16 bits. *ValueLength* deve ser 2.

### **MQTYPE\_INT32**

Um número inteiro assinado de 32 bits. *ValueLength* deve ser 4.

### **MQTYPE\_INT64**

Um número inteiro assinado de 64 bits. *ValueLength* deve ser 8..

### **MQTYPE\_FLOAT32**

Um número de vírgula flutuante de 32 bits.. *ValueLength* deve ser 4.

Nota: esse tipo não é suportado com aplicativos usando IBM COBOL para z/OS.

### **MQTYPE\_FLOAT64**

Um número de vírgula flutuante de 64 bits.. *ValueLength* deve ser 8..

Nota: esse tipo não é suportado com aplicativos usando IBM COBOL para z/OS.

### **MQTYPE\_STRING**

Uma sequência de Caracteres *ValueLength* deve ser zero ou maior ou o valor especial MQVL\_NULL\_TERMINATED.

### **MQTYPE\_NULL**

A propriedade existe mas possui um valor nulo. *ValueLength* deve ser zero.

## ValueLength

Tipo: MQLONG - entrada

O comprimento em bytes do valor da propriedade no parâmetro *Value* . Zero é válido apenas para valores nulos ou para sequências ou sequências de bytes. Zero indica que a propriedade existe, mas que o valor não contém caracteres ou bytes.

O valor deverá ser maior ou igual a zero ou o valor especial a seguir se o parâmetro *Type* tiver MQTYPE\_STRING configurado:

### **MQVL\_NULL\_TERMINATED**

O valor é delimitado pelo primeiro nulo encontrado na cadeia. O nulo não é incluído como parte da cadeia. Este valor é inválido se MQTYPE\_STRING não estiver configurado também.

Nota: O caractere nulo usado para finalizar uma sequência se MQVL\_NULL\_TERMINATED for configurado é nulo do conjunto de caracteres do Valor.

## Value

Tipo: MQBYTExValueComprimento-entrada

O valor da propriedade a ser configurada O buffer deve ser alinhado em um limite apropriado para a natureza dos dados no valor..

Na linguagem de programação C, o parâmetro é declarado como um ponteiro para vazio; o endereço de qualquer tipo de dados pode ser especificado como o parâmetro.

Se *ValueLength* for zero, *Value* não será referido. Nesse caso, o endereço de parâmetro transmitido por programas gravados em C ou assembler System/390 pode ser nulo.

### **CompCode**

Tipo: MQLONG - saída

O código de conclusão; é um dos seguintes:

#### **MQCC\_OK**

Indica conclusão bem-sucedida.

#### **MQCC\_FAILED**

A chamada falhou.

### **Razão**

Tipo: MQLONG - saída

O código de razão que qualifica *CompCode*.

Se *CompCode* for MQCC\_OK:

#### **MQRC\_NONE**

(0, X'000') Nenhuma razão a relatar.

Se *CompCode* for MQCC\_WARNING:

#### **MQRC\_RFH\_FORMAT\_ERROR**

(2421, X'0975') Uma pasta MQRFH2 contendo propriedades não pôde ser analisada.

Se *CompCode* for MQCC\_FAILED:

#### **MQRC\_ADAPTER\_NOT\_AVAILABLE**

(2204, X'089C') Adaptador não disponível.

#### **MQRC\_ADAPTER\_SERV\_LOAD\_ERROR**

(2130, X'852') Não foi possível carregar o módulo de serviço do adaptador.

#### **MQRC\_ASID\_MISMATCH**

(2157, X'86D') ASIDs de início e primárias são diferentes.

#### **MQRC\_BUFFER\_ERROR**

(2004, X'07D4') Parâmetro de valor inválido.

#### **MQRC\_BUFFER\_LENGTH\_ERROR**

(2005, X'07D5') Parâmetro de comprimento do valor inválido.

#### **MQRC\_CALL\_IN\_PROGRESS**

(2219, X'08AB') chamada MQI inserida antes da chamada anterior ser concluída.

#### **MQRC\_HMSG\_ERROR**

(2460, X'099C') Ponteiro do identificador de mensagens inválido.

#### **MQRC\_MSG\_HANDLE\_IN\_USE**

(2499, X'09C3') Identificador de mensagem já em uso.

#### **MQRC\_OPTIONS\_ERROR**

(2046, X'07FE') Opções não válidas ou não consistentes.

#### **MQRC\_PD\_ERROR**

(2482, X'09B2') Estrutura do descritor de propriedade inválida.

#### **MQRC\_PROPERTY\_NAME\_ERROR**

(2442, X'098A') Nome da propriedade inválido

#### **MQRC\_PROPERTY\_TYPE\_ERROR**

(2473, X'09A9') Tipo de dados de propriedade inválido.

**MQRC\_PROP\_NUMBER\_FORMAT\_ERROR**

(2472, X'09A8') Erro de formato numérico encontrado nos dados de valor...

**MQRC\_SMPO\_ERROR**

(2463, X'099F') A estrutura das opções da propriedade de mensagens não é válida...

**MQRC\_SOURCE\_CCSID\_ERROR**

(2111, X'083F') Identificador do conjunto de caracteres codificados do nome da propriedade inválido.

**MQRC\_STORAGE\_NOT\_AVAILABLE**

(2071, X'817') Armazenamento insuficiente disponível.

**MQRC\_UNEXPECTED\_ERROR**

(2195, X'893') Ocorreu um erro inesperado.

Para obter informações detalhadas sobre esses códigos, consulte [Mensagens e códigos de razão](#).

## Chamada C

```
MQSETMP (Hconn, Hmsg, &SetPropOpts, &Name, &PropDesc, Type,
ValueLength, &Value, &CompCode, &Reason);
```

Declare os parâmetros conforme a seguir:

```
MQHCONN  Hconn;          /* Connection handle */
MQHMSG   Hmsg;          /* Message handle */
MQSMPO   SetPropOpts;  /* Options that control the action of MQSETMP */
MQCHARV  Name;         /* Property name */
MQPD     PropDesc;     /* Property descriptor */
MQLONG   Type;         /* Property data type */
MQLONG   ValueLength;  /* Length of property value in Value */
MQBYTE   Value[n];     /* Property value */
MQLONG   CompCode;    /* Completion code */
MQLONG   Reason;      /* Reason code qualifying CompCode */
```

## Chamada COBOL

```
CALL 'MQSETMP' USING HCONN, HMSG, SETMSGOPTS, NAME, PROPDESC, TYPE,
VALUELENGTH, VALUE, COMPCODE, REASON.
```

Declare os parâmetros conforme a seguir:

```
** Connection handle
01 HCONN    PIC S9(9) BINARY.
** Message handle
01 HMSG     PIC S9(18) BINARY.
** Options that control the action of MQSETMP
01 SETMSGOPTS.
   COPY CMQSMPOV.
** Property name
01 NAME
   COPY CMQCHRNV.
** Property descriptor
01 PROPDESC.
   COPY CMQPDV.
** Property data type
01 TYPE     PIC S9(9) BINARY.
** Length of property value in VALUE
01 VALUELENGTH PIC S9(9) BINARY.
** Property value
01 VALUE     PIC X(n).
** Completion code
01 COMPCODE  PIC S9(9) BINARY.
** Reason code qualifying COMPCODE
01 REASON    PIC S9(9) BINARY.
```

## Chamada PL/I

```
call MQSETMP (Hconn, Hmsg, SetPropOpts, Name, PropDesc, Type, ValueLength,  
             Value, CompCode, Reason);
```

Declare os parâmetros conforme a seguir:

```
dcl Hconn      fixed bin(31); /* Connection handle */  
dcl Hmsg      fixed bin(63); /* Message handle */  
dcl SetPropOpts like MQSMP0; /* Options that control the action of MQSETMP */  
dcl Name      like MQCHARV; /* Property name */  
dcl PropDesc  like MQPD; /* Property descriptor */  
dcl Type      fixed bin(31); /* Property data type */  
dcl ValueLength fixed bin(31); /* Length of property value in Value */  
dcl Value     char(n); /* Property value */  
dcl CompCode  fixed bin(31); /* Completion code */  
dcl Reason    fixed bin(31); /* Reason code qualifying CompCode */
```

## Chamada do Assembler de Alto Nível

```
CALL MQSETMP, (HCONN,HMSG,SETMSGHOPTS,NAME,PROPDSC,TYPE,VALUELENGTH,  
              VALUE,COMPCODE,REASON)
```

Declare os parâmetros conforme a seguir:

HCONN	DS	F	Connection handle
HMSG	DS	D	Message handle
SETMSGOPTS	CMQSMP0A	,	Options that control the action of MQSETMP
NAME	CMQCHRVA	,	Property name
PROPDSC	CMQPDA	,	Property descriptor
TYPE	DS	F	Property data type
VALUELENGTH	DS	F	Length of property value in VALUE
VALUE	DS	CL(n)	Property value
COMPCODE	DS	F	Completion code
REASON	DS	F	Reason code qualifying COMPCODE

## MQSTAT-Recuperar informações de status

Use a chamada MQSTAT para recuperar as informações de status. O tipo de informações de status retornadas é determinado pelo valor de Tipo especificado na chamada.

### Sintaxe

MQSTAT (*Hconn*, *Type*, *Stat*, *Compcode*, *Reason*)

### Parâmetros

#### Hconn

Tipo: MQHCONN - entrada

Esta manipulação representa a conexão ao gerenciador de filas. O valor de *Hconn* foi retornado por uma chamada anterior MQCONN ou MQCONNX.

Em z/OS para aplicativos CICS, a chamada MQCONN pode ser omitida e o seguinte valor especificado para *Hconn*:

#### MQHC\_DEF\_HCONN

Manipulação de conexões padrão.

#### tipo

Tipo: MQLONG - entrada

Tipo de informações de status sendo solicitadas. Os valores > válidos são:



**MQSTAT\_TYPE\_ASYNC\_ERROR**

Retornar informações sobre operações de colocação assíncronas anteriores

**MQSTAT\_TYPE\_RECONNECTION**

Retornar informações sobre reconexão. Se a conexão estiver reconectando ou falhou ao reconectar, as informações descrevem a falha que fez com que a conexão começasse a reconectar.

Este valor é válido apenas para conexões do cliente. Para outros tipos de conexão, a chamada falha com o código de razão **MQRC\_ENVIRONMENT\_ERROR**

**MQSTAT\_TYPE\_RECONNECTION\_ERROR**

Retornar informações sobre uma falha anterior relacionada à reconexão. Se a conexão falhou ao se reconectar, as informações descrevem a falha que causou a falha da reconexão.

Este valor é válido apenas para conexões do cliente. Para outros tipos de conexão, a chamada falha com código de razão **MQRC\_ENVIRONMENT\_ERROR**.

**stat**

Tipo: MQSTS-entrada/saída

Estrutura de informações de status Consulte [“MQSTS-Estrutura de relatório de status.”](#) na página 613 para obter detalhes.

**CompCode**

Tipo: MQLONG - saída

O código de conclusão; é um dos seguintes:

**MQCC\_OK**

Indica conclusão bem-sucedida.

**MQCC\_FAILED**

A chamada falhou.

**Razão**

Tipo: MQLONG - saída

O código de razão que qualifica *CompCode*.

Se *CompCode* for MQCC\_OK:

**MQRC\_NONE**

(0, X'000') Nenhuma razão a relatar.

Se *CompCode* for MQCC\_FAILED:

**MQRC\_API\_EXIT\_ERROR**

(2374, X'946 ') A saída da API falhou

**MQRC\_API\_EXIT\_LOAD\_ERROR**

(2183, X'887') Não foi possível carregar a saída de API.

**MQRC\_CALL\_IN\_PROGRESS**

(2219, X'8AB') Chamada MQI inserida antes da chamada anterior concluída.

**MQRC\_CONNECTION\_BROKEN**

(2009, X'7D9') Conexão ao gerenciador de filas perdida.

**MQRC\_CONNECTION\_STOPPING**

(2203, X'89B') Conexão sendo encerrada.

**MQRC\_FUNCTION\_NOT\_SUPPORTED**

(2298, X'8FA') A função solicitada não está disponível no ambiente atual.

**MQRC\_HCONN\_ERROR**

(2018, X'7E2') Manipulação de conexões não válida.

**MQRC\_Q\_MGR\_STOPPING**

(2162, X'872')-Gerenciador de filas parando

**MQRC\_RESOURCE\_PROBLEM**

(2102, X'836') Recursos insuficientes do sistema disponíveis.

**MQRC\_STAT\_TYPE\_ERROR**

(2430, X'97E' Erro com tipo MQSTAT

**MQRC\_STORAGE\_NOT\_AVAILABLE**

(2071, X'817') Armazenamento insuficiente disponível.

**MQRC\_STS\_ERROR**

(2426, X'97A') Erro com a estrutura MQSTS

**MQRC\_UNEXPECTED\_ERROR**

(2195, X'893') Ocorreu um erro inesperado.

Para obter informações detalhadas sobre esses códigos, consulte [Mensagens e códigos de razão](#).

**Observações de Uso**

1. Uma chamada para MQSTAT especificando um tipo de MQSTAT\_TYPE\_ASYNC\_ERROR retorna informações sobre as operações assíncronas MQPUT e MQPUT1 anteriores. A estrutura MQSTS transmitida de volta no retorno da chamada MQSTAT contém as primeiras informações de aviso assíncronas ou de erro registradas para essa conexão. Se erros ou avisos adicionais seguirem o primeiro, eles normalmente não alterarão esses valores. No entanto, se ocorrer um erro com um código de conclusão de MQCC\_WARNING, uma falha subsequente com um código de conclusão de MQCC\_FAILED será retornada.
2. Se nenhum erro tiver ocorrido desde que a conexão foi estabelecida ou desde a última chamada para MQSTAT, um CompCode de MQCC\_OK e um Motivo de MQRC\_NONE serão retornados na estrutura MQSTS.
3. As contagens do número de chamadas assíncronas que foram processadas sob a manipulação de conexões são retornadas por meio de três campos do contador; PutSuccessCount, PutWarningCount e PutFailureCount. Esses contadores são incrementados pelo gerenciador de filas toda vez que uma operação assíncrona é processada com êxito, possui um aviso ou falha (observe que, para fins de contabilidade, uma colocação em uma lista de distribuição conta uma vez por fila de destino em vez de uma vez por lista de distribuição). Um contador não é incrementado além do valor máximo positivo AMQ\_LONG\_MAX.
4. Uma chamada bem-sucedida para MQSTAT resulta em quaisquer informações de erro ou contagens anteriores sendo reconfiguradas.
5. O comportamento de MQSTAT depende do valor do parâmetro **MQSTAT Type** fornecido.
6. **MQSTAT\_TYPE\_ASYNC\_ERROR**
  - a. Uma chamada para MQSTAT especificando um tipo de MQSTAT\_TYPE\_ASYNC\_ERROR retorna informações sobre as operações assíncronas MQPUT e MQPUT1 anteriores. A estrutura MQSTS transmitida de volta no retorno da chamada MQSTAT contém as primeiras informações de aviso assíncronas ou de erro registradas para essa conexão. Se erros ou avisos adicionais seguirem o primeiro, eles normalmente não alterarão esses valores. No entanto, se ocorrer um erro com um código de conclusão de MQCC\_WARNING, uma falha subsequente com um código de conclusão de MQCC\_FAILED será retornada.
  - b. Se nenhum erro tiver ocorrido desde que a conexão foi estabelecida ou desde a última chamada para MQSTAT, um CompCode de MQCC\_OK e um Motivo de MQRC\_NONE serão retornados na estrutura MQSTS.
  - c. As contagens do número de chamadas assíncronas que foram processadas sob a manipulação de conexões são retornadas por meio de três campos do contador; PutSuccessCount, PutWarningCount e PutFailureCount. Esses contadores são incrementados pelo gerenciador de filas toda vez que uma operação assíncrona é processada com êxito, possui um aviso ou falha (observe que, para fins de contabilidade, uma colocação em uma lista de distribuição conta uma vez por fila de destino em vez de uma vez por lista de distribuição). Um contador não é incrementado além do valor máximo positivo AMQ\_LONG\_MAX.

- d. Uma chamada bem-sucedida para MQSTAT resulta em quaisquer informações de erro ou contagens anteriores sendo reconfiguradas

### MQSTAT\_TYPE\_RECONNECTION

Suponha que você chame MQSTAT com Type configurado como MQSTAT\_TYPE\_RECONNECTION dentro de um manipulador de eventos durante a reconexão. Considere estes exemplos.

#### O cliente está tentando reconectar ou falhou ao reconectar.

CompCode na estrutura MQSTS é MQCC\_FAILED e Reason pode ser MQRC\_CONNECTION\_BROKEN ou MQRC\_Q\_MGR QUIESCING. ObjectType é MQOT\_Q\_MGR, ObjectName é o nome do gerenciador de filas e ObjectQMgrName está em branco..

#### O cliente concluiu a reconexão com êxito ou nunca foi desconectado.

CompCode na estrutura MQSTS é MQCC\_OK e o Reason é MQRC\_NONE

Chamadas subsequentes para MQSTAT retornam os mesmos resultados.

### MQSTAT\_TYPE\_RECONNECTION\_ERROR

Suponha que você chame MQSTAT com Type configurado como MQSTAT\_TYPE\_RECONNECTION\_ERROR em resposta ao recebimento de MQRC\_RECONNECT\_FAILED para uma chamada MQI. Considere estes exemplos.

#### Ocorreu uma falha de autorização quando uma fila estava sendo reaberta durante a reconexão com um gerenciador de filas diferente

CompCode na estrutura MQSTS é MQCC\_FAILED e Reason é o motivo pelo qual a reconexão falhou, como MQRC\_NOT\_AUTHORIZED. ObjectType é o tipo de objeto que causou o problema, como MQOT\_QUEUE, ObjectName é o nome da fila e ObjectQMgrName o nome do gerenciador de filas que possui a fila.

#### Ocorreu um erro de conexão do soquete durante a reconexão

CompCode na estrutura MQSTS é MQCC\_FAILED e Reason é o motivo pelo qual a reconexão falhou, como MQRC\_HOST\_NOT\_AVAILABLE. ObjectType é MQOT\_Q\_MGR, ObjectName é o nome do gerenciador de filas e ObjectQMgrName está em branco..

Chamadas subsequentes para MQSTAT retornam os mesmos resultados.

## Chamada C

```
MQSTAT (Hconn, StatType, &Stat, &CompCode, &Reason);
```

Declare os parâmetros conforme a seguir:

```
MQHCONN Hconn;          /* Connection Handle */
MQLONG StatType;        /* Status type */
MQSTS Stat;             /* Status information structure */
MQLONG CompCode;        /* Completion code */
MQLONG Reason;          /* Reason code qualifying CompCode */
```

## Chamada COBOL

```
CALL 'MQSTAT' USING HCONN, STATTYPE, STAT, COMPCODE, REASON.
```

Declare os parâmetros conforme a seguir:

```
**      Connection handle
01     HCONN      PIC S9(9)      BINARY.
**      Status type
01     STATTYPE  PIC S9(9)      BINARY.
**      Status information
01     STAT.
**      COPY CMQSTSV.
**      Completion code
```

```

01    COMPCODE    PIC S9(9)      BINARY.
**   Reason code qualifying COMPCODE
01    REASON      PIC S9(9)      BINARY.

```

## Chamada PL/I

```
call MQSTAT (Hconn, StatType, Stat, Compcode, Reason);
```

Declare os parâmetros conforme a seguir:

```

dcl Hconn          fixed bin(31); /* Connection handle */
dcl StatType      fixed bin(31); /* Status type */
dcl Stat          like MQSTS;    /* Status information structure */
dcl CompCode     fixed bin(31); /* Completion code */
dcl Reason       fixed bin(31); /* Reason code qualifying CompCode */

```

## System/390 Chamada do Assembler

```
CALL MQSTAT, (HCONN, STATTYPE, STAT, COMPCODE, REASON)
```

Declare os parâmetros conforme a seguir:

HCONN	DS	F	Connection handle
STATTYPE	DS	F	Status type
STAT	CMQSTSA,		Status information structure
COMPCODE	DS	F	Completion code
REASON	DS	F	Reason code qualifying COMPCODE

## MQSUB - Assinatura do registro

Use a chamada MQSUB para registrar a assinatura de aplicativos para um determinado tópico.

### Sintaxe

```
MQSUB (Hconn, SubDesc, Hobj, Hsub, Compcode, Reason)
```

### Parâmetros

#### Hconn

Tipo: MQHCONN - entrada

Esta manipulação representa a conexão ao gerenciador de filas. O valor de *Hconn* foi retornado por uma chamada anterior MQCONN ou MQCONNX.

Em z/OS para aplicativos CICS, a chamada MQCONN pode ser omitida e o seguinte valor especificado para *Hconn*:

#### MQHC\_DEF\_HCONN

Manipulação de conexões padrão.

#### SubDesc

Tipo: MQSD - entrada/saída

Esta é uma estrutura que identifica o objeto em uso que está sendo registrado pelo aplicativo. Consulte a [“MQSD - Descritor de Assinatura” na página 588](#) para obter mais informações.

#### Hobj

Tipo: MQHOBJ - entrada/saída

Esta manipulação representa o acesso que foi estabelecido para obter as mensagens enviadas a essa assinatura. Essas mensagens podem ser armazenadas em uma fila específica ou o gerenciador de filas pode gerenciar seus armazenamentos sem usar uma fila específica.

Para usar uma fila específica, você deve associá-la à assinatura quando a assinatura for criada. Isso pode ser feito de duas maneiras:

- Usando o comando DEFINE SUB MQSC e fornecendo a esse comando o nome de um objeto de fila.
- Fornecendo esta manipulação ao chamar MQSUB com MQSO\_CREATE

Se esta manipulação for fornecida como um parâmetro de entrada na chamada, ela deve ser uma manipulação de objeto válida retornada a partir de uma chamada MQOPEN anterior de uma fila que usa, pelo menos, uma das seguintes opções:

- MQOO\_INPUT\_\*
- MQOO\_BROWSE
- MQOO\_OUTPUT (se a fila for uma fila remota)

Se esse não for o caso, a chamada falhará com MQRC\_HOBJ\_ERROR. Não pode ser uma manipulação de objetos para uma fila de alias que seja resolvida em um objeto do tópico. Nesse caso, a chamada falhará com MQRC\_HOBJ\_ERROR.

Se o gerenciador de filas gerenciar o armazenamento de mensagens enviadas a esta assinatura, ele deve ser configurado ao criar a assinatura, usando a opção MQSO\_MANAGED. O gerenciador de filas então retorna esta manipulação como um parâmetro de saída na chamada. A manipulação que é retornada é conhecida como uma manipulação gerenciada. Se MQHO\_NONE for especificado, mas MQSO\_MANAGED não for especificado, a chamada falhará com MQRC\_HOBJ\_ERROR.

Quando uma manipulação gerenciada for retornada pelo gerenciador de filas, será possível usá-la em uma chamada MQGET ou MQCB com ou sem as opções de procura, em uma chamada MQINQ ou no MQCLOSE. Você não pode usá-la em MQPUT, MQSUB, MQSET; a tentativa de tal procedimento falha com MQRC\_NOT\_OPEN\_FOR\_OUTPUT, MQRC\_HOBJ\_ERROR, or MQRC\_NOT\_OPEN\_FOR\_SET.

Se esta assinatura estiver sendo continuada usando a opção MQSO\_RESUME na estrutura MQSD, a manipulação pode ser retornada ao aplicativo neste parâmetro, configurando MQSO\_MANAGED para MQHO\_NONE. É possível fazer isso se a assinatura estiver usando a manipulação gerenciada, ou não, e pode ser útil fornecer assinaturas criadas usando DEFINE SUB com a manipulação para a fila de assinatura definida nesse comando. No caso em que uma assinatura criada administrativamente está sendo continuada, a fila é aberta com MQOO\_INPUT\_AS\_Q\_DEF e MQOO\_BROWSE. Se você precisar especificar outras opções, o aplicativo deve abrir a fila de assinaturas explicitamente e fornecer a manipulação de objetos na chamada. Se ocorrer um problema ao abrir a fila, a chamada falhará com MQRC\_INVALID\_DESTINATION. Se o *Hobj* for fornecido, ele deverá ser equivalente ao *Hobj* na chamada MQSUB original. Isso significa que se uma manipulação de objetos retornada de uma chamada MQOPEN estiver sendo fornecida, a manipulação deve estar na mesma fila da anteriormente usada. Se não for a mesma fila, a chamada falhará com MQRC\_HOBJ\_ERROR.

Se essa assinatura estiver sendo alterada usando a opção MQSO\_ALTER na estrutura MQSD, um *Hobj* diferente poderá ser fornecido. Qualquer publicação que tenha sido entregue para a fila e anteriormente identificada por meio desse parâmetro permanecerá nessa fila e será responsabilidade do aplicativo recuperar essas mensagens se o parâmetro **Hobj** agora representar uma fila diferente.

Tabela 555. Usando o <i>hobj</i> com várias opções de assinatura		
Opções	<i>Hobj</i>	Descrição
MQSO_CREATE + MQSO_MANAGED	Ignorado na entrada	Cria uma assinatura com o armazenamento das mensagens gerenciadas pelo gerenciador de fila

Tabela 555. Usando o <i>hobj</i> com várias opções de assinatura (continuação)		
Opções	Hobj	Descrição
MQSO_CREATE	Uma manipulação de objetos válida	Cria uma assinatura que fornece uma fila específica como destino para as mensagens.
MQSO_RESUME	MQHO_NONE	Continua uma assinatura anteriormente criada independentemente de ter sido gerenciada, ou não, e faz com que o gerenciador de filas retorne a manipulação de objetos para uso pelo aplicativo.
MQSO_RESUME	Uma manipulação de objetos válida, correspondente	Continua uma assinatura anteriormente criada que usa uma fila específica como destino para as mensagens e usa uma manipulação de objetos com opções abertas específicas.
MQSO_ALTER + MQSO_MANAGED	MQHO_NONE	Altera a assinatura existente que estava usando anteriormente uma fila específica; portanto, agora é uma assinatura gerenciada. Não é possível mudar a classe de destino (gerenciada ou não).
MQSO_ALTER	Uma manipulação de objetos válida	Altera uma assinatura existente, independentemente de ter sido gerenciada, ou não; portanto, agora usa uma fila específica. Quando a opção MQSO_MANAGED não for usada, a fila fornecida pode ser mudada, mas a classe de destino (gerenciada ou não) não pode ser mudada.

Independentemente de ter sido fornecido ou retornado, *Hobj* deve ser especificado em chamadas MQGET ou MQCB subsequentes que desejam receber as mensagens de publicação enviadas para esta assinatura.

A manipulação *Hobj* não é mais válida quando a chamada MQCLOSE é emitida nela ou quando a unidade de processamento que define o escopo da manipulação termina (até que o aplicativo seja desconectado). O escopo de manipulação de objetos retornado é igual ao da manipulação de conexões especificada na chamada. Consulte [Hconn \(MQHCONN\)](#) - saída para obter informações sobre o escopo de manipulação. Um MQCLOSE da manipulação *Hobj* não afeta a manipulação *Hsub*.

### Hsub

Tipo: MQHOBJ - saída

Esta manipulação representa a assinatura que foi feita. Ela pode ser usada para duas operações adicionais:

- Ela pode ser usada em uma chamada MQSUBRQ subsequente para solicitar que as publicações sejam enviadas quando a opção MQSO\_PUBLICATIONS\_ON\_REQUEST tiver sido usada ao fazer a assinatura.
- Ela pode ser usada em uma chamada MQCLOSE subsequente para remover a assinatura que foi feita. A manipulação *Hsub* deixa de ser válida quando a chamada MQCLOSE é emitida ou quando a

unidade de processamento que define o escopo da manipulação termina. O escopo de manipulação de objetos retornado é igual ao da manipulação de conexões especificada na chamada. Um MQCLOSE da manipulação *Hsub* não afeta a manipulação *Hobj*.

Esta manipulação não pode ser passada para uma chamada MQGET ou MQCB. Deve-se usar o parâmetro **Hobj**. Não é possível usar essa manipulação em nenhuma chamada IBM MQ diferente de MQCLOSE ou MQSUBRQ. Transmitir essa manipulação para qualquer outra chamada IBM MQ resulta em MQRC\_HOBY\_ERROR.

### CompCode

Tipo: MQLONG - saída

O código de conclusão; é um dos seguintes:

#### **MQCC\_OK**

Conclusão bem-sucedida

#### **MQCC\_WARNING**

Aviso (conclusão parcial)

#### **MQCC\_FAILED**

Chamada com falha

### Razão

Tipo: MQLONG - saída

O código de razão que qualifica *CompCode*.

Se *CompCode* for MQCC\_OK, o código de razão será o seguinte:

#### **MQRC\_NONE**

(0, X'000') Nenhuma razão a relatar.

Se *CompCode* for MQCC\_FAILED, o código de razão será um dos seguintes:

#### **MQRC\_CLUSTER\_RESOLUTION\_ERROR**

(2189, X'88D') Falha na resolução do nome do cluster.

#### **MQRC\_DURABILITY\_NOT\_ALLOWED**

2436 (X'0984') Uma chamada MQSUB que usa a opção MQSO\_DURABLE falhou.

#### **MQRC\_FUNCTION\_NOT\_SUPPORTED**

2298 (X'08FA') A função solicitada não está disponível no ambiente atual.

#### **MQRC\_HOBY\_ERROR**

2019 (X'07E3') Manipulação de objetos Hobj não válida.

#### **MQRC\_IDENTITY\_MISMATCH**

2434 (X'0982') O nome da assinatura corresponde à assinatura existente.

#### **MQRC\_NOT\_AUTHORIZED**

2035 (X'07F3') O usuário não está autorizado a executar a operação.

#### **MQRC\_NO\_SUBSCRIPTION**

2428 (X'097C') O nome de assinatura identificado não existe.

#### **MQRC\_OBJECT\_STRING\_ERROR**

2441 (X'0989') Campo Objectstring não válido.

#### **MQRC\_OPTIONS\_ERROR**

2046 (X'07FE') O parâmetro ou campo Options contém opções que não são válidas ou uma combinação das opções que não é válida.

#### **MQRC\_Q\_MGR QUIESCING**

2161 (X'0871') Quiesce do gerenciador de filas.

#### **MQRC\_RECONNECT\_Q\_MGR\_REQD**

2555 (X'09FB'X) A opção MQCNO\_RECONNECT\_Q\_MGR é necessária.

**MQRC\_RETAINED\_MSG\_Q\_ERROR**

2525 (X'09DD') As publicações retidas que existem para a sequência de tópicos inscritas não podem ser recuperadas.

**MQRC\_RETAINED\_NOT\_DELIVERED**

2526 (X'09DE') As publicações retidas, que existem para a sequência de tópicos inscrita, não podem ser entregues para a fila de destino da assinatura e não podem ser entregues para a fila de devoluções.

**MQRC\_SD\_ERROR**

2424 (X'0978') Descritor de assinatura (MQSD) não válido.

**MQRC\_SELECTION\_NOT\_AVAILABLE**

2551 (X'09F7') A sequência de seleção não segue a sintaxe do seletor IBM MQ e nenhum provedor de seleção de mensagem estendido estava disponível.

**MQRC\_SELECTION\_STRING\_ERROR**

2519 (X'09D7') A sequência de seleção deve ser especificada conforme descrito na documentação da estrutura MQCHARV.

**MQRC\_SELECTOR\_SYNTAX\_ERROR**

2459 (X'099B') Uma chamada MQOPEN, MQPUT1 ou MQSUB foi emitida, mas foi especificada uma sequência de seleção que continha um erro de sintaxe.

**MQRC\_SUB\_USER\_DATA\_ERROR**

2431 (X'097F') Campo SubUserData não válido.

**MQRC\_SUB\_NAME\_ERROR**

2440 (X'0988') Campo SubName não válido.

**MQRC\_SUB\_ALREADY\_EXISTS**

2432 (X'0980') A assinatura já existe.

**MQRC\_SUB\_USER\_DATA\_ERROR**

2431 (X'097F') Campo SubUserData não válido.

**MQRC\_TOPIC\_STRING\_ERROR**

2425 (X'0979') A sequência de tópicos não é válida.

**MQRC\_UNKNOWN\_OBJECT\_NAME**

2085 (X'0825') Não é possível localizar o objeto identificado no campo ObjectName de MQSD.

**MQRC\_SUB\_JOIN\_NOT\_ALTERABLE**

29440 (X'7300') O modo de compartilhamento de assinaturas é incompatível com a assinatura existente. Esse erro poderia ser retornado ao tentar continuar uma assinatura compartilhada do JMS 2.0 em um aplicativo não JMS.

Para obter informações detalhadas sobre esses códigos, consulte [Mensagens e códigos de razão](#).

## Observações de Uso

- A assinatura é feita em um tópico, nomeado usando o nome abreviado de um objeto de tópico predefinido, o nome completo da sequência de tópicos ou é formada pela concatenação das duas partes. Consulte a descrição de *ObjectName* e *ObjectString* em [“MQSD - Descritor de Assinatura”](#) na página 588.
- O gerenciador de filas executa verificações de segurança quando uma chamada MQSUB é emitida, para verificar se o identificador do usuário sob o qual o aplicativo é executado possui o nível apropriado de autoridade antes que o acesso seja permitido. O objeto do tópico apropriado está localizado na hierarquia do tópico e uma verificação de autoridade é feita neste objeto do tópico para assegurar se a autoridade para subscrever está configurada. Se a opção MQSO\_MANAGED não for usada, será feita uma verificação de autoridade na fila de destino para assegurar se a autoridade para a saída está configurada. Se a opção MQSO\_MANAGED for usada, nenhuma verificação de autoridade será feita na fila gerenciada para saída ou acesso de consulta.
- Se você não fornecer um Hobj como entrada, a chamada MQSUB alocará duas manipulações, uma manipulação de objetos (Hobj) e uma manipulação de assinatura (Hsub).



- O Hobj retornado na chamada MQSUB quando a opção MQSO\_MANAGED é usada pode ser consultado para localizar atributos como o limite de Restauração e o nome do enfileiramento de restauração Excessiva. Também é possível consultar o nome de fila gerenciada, mas você não deve tentar abrir diretamente esta fila.
  - As assinaturas podem ser agrupadas permitindo que apenas uma única publicação seja entregue ao grupo de assinaturas mesmo quando mais de um grupo tiver correspondido à publicação. As assinaturas são agrupadas usando a opção MQSO\_GROUP\_SUB e para agrupar as assinaturas elas devem estar
    - usando a mesma fila nomeada (que não está usando a opção MQSO\_MANAGED) no mesmo gerenciador de filas - representada pelo parâmetro Hobj na chamada MQSUB
    - compartilhar o mesmo SubCorrelId
    - ser do mesmo Subnível
- Esses atributos definem o conjunto de assinaturas consideradas no grupo e também são os atributos que não podem ser alterados, se uma assinatura for agrupada. A mudança do Subnível resulta em MQRC\_SUBLEVEL\_NOT\_ALTERABLE e a mudança de quaisquer outros resultados (que podem ser alterados, se uma assinatura não for agrupada) em MQRC\_GROUPING\_NOT\_ALTERABLE.
- A conclusão bem-sucedida da chamada MQSUB não significa que a ação tenha sido concluída. Para verificar se essa chamada foi concluída, consulte a etapa [DEFINE SUB](#) em [Verificando se os comandos assíncronos para redes distribuídas foram concluídos](#).
  - Os campos no MQSD são preenchidos no retorno de uma chamada MQSUB que usa a opção MQSO\_RESUME. O MQSD retornado pode ser passado diretamente para uma chamada MQSUB que usa a opção MQSO\_ALTER com quaisquer mudanças que precisa fazer na assinatura aplicada ao MQSD. Alguns campos possuem considerações especiais conforme observado na tabela.

<i>Tabela 556. Considerações especiais para campos no MQSD</i>	
<b>Nome do campo em MQSD</b>	<b>Considerações Especiais</b>
Opções de Acesso ou Criação	Algumas das opções podem ser reconfiguradas no retorno da chamada MQSUB. Se, em seguida, você reutilizar o MQSD em uma chamada MQSUB, a opção necessária deverá ser configurada explicitamente.
Opções de Durabilidade, opções de Destino, opções de Registro & opções de Curinga	Essas opções são configuradas conforme apropriado
Opções de publicação	Essas opções são configuradas conforme apropriado, exceto para MQSO_NEW_PUBLICATIONS_ONLY que é aplicável apenas para MQSO_CREATE.
Outras Opções	Essas opções ficam inalteradas no retorno de uma chamada MQSUB. Elas controlam como a chamada de API é emitida e não são armazenadas com a assinatura. Elas devem ser configuradas conforme necessário em qualquer chamada MQSUB subsequente reutilizando o MQSD.
ObjectName	Este campo de entrada apenas fica inalterado no retorno de uma chamada MQSUB.
ObjectString	Este campo de entrada apenas fica inalterado no retorno de uma chamada MQSUB. O nome do Tópico integral usado será retornado no campo <i>ResObjectString</i> , se for fornecido um buffer.

Tabela 556. Considerações especiais para campos no MQSD (continuação)

Nome do campo em MQSD	Considerações Especiais
AlternateUserId e AlternateSecurityId	Esses campos de entrada apenas ficam inalterados no retorno de uma chamada MQSUB. Elas controlam como a chamada de API é emitida e não são armazenadas com a assinatura. Elas devem ser configuradas conforme necessário em qualquer chamada MQSUB subsequente reutilizando o MQSD.
SubExpiry	No retorno de uma chamada MQSUB usando a opção MQSO_RESUME, esse campo é configurado para a expiração original da assinatura e não o tempo de expiração restante. Se, em seguida, você reutilizar o MQSD em uma chamada MQSUB usando a opção MQSO_ALTER, reconfigure a expiração da assinatura para iniciar a contagem regressiva novamente.
SubName	Este campo é um campo de entrada em uma chamada MQSUB e não é alterado na saída.
SubUserData e SelectionString	<p>Esses campos de comprimento variável serão retornados na saída de uma chamada MQSUB usando a opção MQSO_RESUME, se for fornecido um buffer e também um comprimento de buffer positivo em <i>VSBuFSIZE</i>. Se nenhum buffer for fornecido, apenas o comprimento será retornado no campo <i>VSLength</i> do MQCHARV. Se o buffer fornecido for menor que o espaço necessário para retornar o campo, apenas <i>VSBuFSIZE</i> bytes serão retornados no buffer fornecido.</p> <p>Se, em seguida, você reutilizar o MQSD em uma chamada MQSUB usando a opção MQSO_ALTER e um buffer não for fornecido, mas um <i>VSLength</i> diferente de zero for fornecido, e se esse comprimento corresponder ao comprimento existente do campo, nenhuma mudança será feita no campo.</p>
SubCorrelId e PubAccountingToken	<p>Se você não usar MQSO_SET_CORREL_ID, o <i>SubCorrelId</i> será gerado pelo gerenciador de filas. Se você não usar MQSO_SET_IDENTITY_CONTEXT, o <i>PubAccountingToken</i> será gerado pelo gerenciador de filas.</p> <p>Esses campos são retornados no MQSD a partir de uma chamada MQSUB usando a opção MQSO_RESUME. Se eles forem gerados pelo gerenciador de filas, o valor gerado será retornado em uma chamada MQSUB usando a opção MQSO_CREATE ou MQSO_ALTER.</p>
PubPriority, SubLevel & PubApplIdentityData	Esses campos são retornados no MQSD.
ResObjectString	Este campo de saída apenas será retornado no MQSD, se for fornecido um buffer.

## Chamada C

```
MQSUB (Hconn, &SubDesc, &Hobj, &Hsub, &CompCode, &Reason)
```

Declare os parâmetros conforme a seguir:

```
MQHCONN Hconn; /* Connection handle */
MQSD SubDesc; /* Subscription descriptor */
MQHOBJ Hobj; /* Object handle */
MQHOBJ Hsub; /* Subscription handle */
MQLONG CompCode; /* Completion code */
MQLONG Reason; /* Reason code qualifying CompCode */
```

## Chamada COBOL

```
CALL 'MQSUB' USING HCONN, SUBDESC, HOBJ, HSUB, COMPCODE, REASON.
```

Declare os parâmetros conforme a seguir:

```
** Connection handle
01 HCONN PIC S9(9) BINARY.
** Subscription descriptor
01 SUBDESC.
COPY CMQSDV.
** Object handle
01 HOBJ PIC S9(9) BINARY.
** Subscription handle
01 HSUB PIC S9(9) BINARY.
** Completion code
01 COMPCODE PIC S9(9) BINARY.
** Reason code qualifying COMPCODE
01 REASON PIC S9(9) BINARY.
```

## Chamada PL/I

```
call MQSUB (Hconn, SubDesc, Hobj, Hsub, CompCode, Reason)
```

Declare os parâmetros conforme a seguir:

```
dcl Hconn fixed bin(31); /* Connection handle */
dcl SubDesc like MQSD; /* Subscription descriptor */
dcl Hobj fixed bin(31); /* Object handle */
dcl Hsub fixed bin(31); /* Subscription handle */
dcl CompCode fixed bin(31); /* Completion code */
dcl Reason fixed bin(31); /* Reason code qualifying CompCode */
```

## Chamada do Assembler de Alto Nível

```
CALL MQSUB, (HCONN, SUBDESC, HOBJ, HSUB, COMPCODE, REASON)
```

Declare os parâmetros conforme a seguir:

```
HCONN DS F Connection handle
SUBDESC CMQSDA , Subscription descriptor
HOBJ DS F Object handle
HSUB DS F Subscription handle
COMPCODE DS F Completion code
REASON DS F Reason code qualifying COMPCODE
```

## **MQSUBRQ-Pedido de assinatura**

Use a chamada MQSUBRQ para fazer uma solicitação para a publicação retida, quando o assinante tiver sido registrado com MQSO\_PUBLICATIONS\_ON\_REQUEST

### **Sintaxe**

MQSUBRQ (*Hconn*, *Hsub*, *Ação*, *SubRqOpts*, *Compcode*, *Motivo*)..

### **Parâmetros**

#### **Hconn**

Tipo: MQHCONN - entrada

Esta manipulação representa a conexão ao gerenciador de filas. O valor de *Hconn* foi retornado por uma chamada anterior MQCONN ou MQCONNX.

Em z/OS para aplicativos CICS, a chamada MQCONN pode ser omitida e o seguinte valor especificado para *Hconn*:

#### **MQHC\_DEF\_HCONN**

Manipulação de conexões padrão.

#### **Hsub**

Tipo: MQHOBJ - entrada

Esse identificador representa a assinatura para a qual uma atualização deve ser solicitada. O valor de *Hsub* foi retornado de uma chamada MQSUB anterior.

#### **Ação**

Tipo: MQLONG - entrada

Esse parâmetro controla a ação específica que está sendo solicitada na assinatura.. O seguinte valor deve ser especificado:

#### **MQSR\_ACTION\_PUBLICATION**

Esta ação solicita que uma publicação de atualização seja enviada para o tópico especificado. Ele pode ser usado somente se o assinante especificou a opção MQSO\_PUBLICATIONS\_ON\_REQUEST na chamada MQSUB quando ele fez a assinatura. Se o gerenciador de filas tiver uma publicação retida para o tópico, isto será enviado ao assinante. Se não, a chamada falhará.. Se um aplicativo receber uma publicação que foi retida, isso será indicado pela propriedade da mensagem MQIsRetained dessa publicação.

Como o tópico na assinatura existente representada pelo parâmetro *Hsub* pode conter curingas, o assinante pode receber várias publicações retidas.

#### **SubRqOpts**

Tipo: MQSRO-entrada/saída

Essas opções controlam a ação de MQSUBRQ, consulte [“MQSRO-Opções de solicitação de assinatura”](#) na página 611 para obter detalhes..

Se nenhuma opção for necessária, os programas gravados no assembler C ou S/390 poderão especificar um endereço de parâmetro nulo em vez de especificar o endereço de uma estrutura MQSRO.

#### **CompCode**

Tipo: MQLONG - saída

O código de conclusão; é um dos seguintes:

#### **MQCC\_OK**

Conclusão bem-sucedida

#### **MQCC\_WARNING**

Aviso (conclusão parcial)

**MQCC\_FAILED**

Chamada com falha

**Razão**

Tipo: MQLONG - saída

O código de razão que qualifica *CompCode*.

Se *CompCode* for MQCC\_OK:

**MQRC\_NONE**

(0, X'000') Nenhuma razão a relatar.

Se *CompCode* for MQCC\_FAILED:

**MQRC\_FUNCTION\_NOT\_SUPPORTED**

2298 (X'08FA') A função solicitada não está disponível no ambiente atual.

**MQRC\_NO\_RETAINED\_MSG**

2437 (X'0985 ') Não há publicações armazenadas atualmente para este tópico.

**MQRC\_OPTIONS\_ERROR**

2046 (X'07FE') O parâmetro ou campo Options contém opções que não são válidas ou uma combinação das opções que não é válida.

**MQRC\_Q\_MGR QUIESCING**

2161 (X'0871') Quiesce do gerenciador de filas.

**MQRC\_SRO\_ERROR**

2438 (X'0986 ') Na chamada MQSUBRQ, o MQSRO Subscription Request Options não é válido.

**MQRC\_RETAINED\_MSG\_Q\_ERROR**

2525 (X'09DD') As publicações retidas que existem para a sequência de tópicos inscritas não podem ser recuperadas.

**MQRC\_RETAINED\_NOT\_DELIVERED**

2526 (X'09DE') As publicações retidas, que existem para a sequência de tópicos inscrita, não podem ser entregues para a fila de destino da assinatura e não podem ser entregues para a fila de devoluções.

Para obter informações detalhadas sobre esses códigos, consulte [Mensagens e códigos de razão](#).

**Observações de Uso**

As notas de uso a seguir se aplicam ao uso do código de Ação MQSR\_ACTION\_PUBLICATION:

1. Se esse verbo for concluído com êxito, as publicações retidas correspondentes à assinatura especificada foram enviadas para a assinatura e podem ser recebidas usando MQGET ou MQCB usando o Hobj retornado no verbo MQSUB original que criou a assinatura..
2. Se o tópico subscrito pelo verbo MQSUB original que criou a assinatura continha um curinga, mais de uma publicação retida poderá ser enviada O número de publicações enviadas como resultado dessa chamada é registrado no campo NumPubs na estrutura Opts SubRq.
3. Se esse verbo for concluído com um código de razão de MQRC\_NO\_RETAINED\_MSG, não haverá publicações retidas atualmente para o tópico especificado. #
4. Se esse verbo for concluído com um código de razão de MQRC\_RETAINED\_MSG\_Q\_ERROR ou MQRC\_RETAINED\_NOT\_ENTREGUE, então há publicações retidas atualmente para o tópico especificado, mas ocorreu um erro que significava que elas não puderam ser entregues..
5. O aplicativo deve ter uma assinatura atual para o tópico antes que ele possa fazer esta chamada. Se a assinatura tiver sido feita em uma instância anterior do aplicativo e um identificador válido para a assinatura não estiver disponível, o aplicativo deverá primeiro chamar MQSUB com a opção MQSO\_RESUME para obter um identificador para ele para uso nessa chamada
6. As publicações são enviadas para o destino registrado para uso com a assinatura atual deste aplicativo. Se as publicações tiverem que ser enviadas para outro lugar, a assinatura deverá primeiro ser alterada usando a chamada MQSUB com a opção MQSO\_ALTER

## Chamada C

```
MQSUB (Hconn, Hsub, Action, &SubRqOpts, &CompCode, &Reason)
```

Declare os parâmetros conforme a seguir:

```
MQHCONN Hconn; /* Connection handle */
MQHOBJ Hsub; /* Subscription handle */
MQLONG Action; /* Action requested by MQSUBRQ */
MQSRO SubRqOpts; /* Options that control the action of MQSUBRQ */
MQLONG CompCode; /* Completion code */
MQLONG Reason; /* Reason code qualifying CompCode */
```

## Chamada COBOL

```
CALL 'MQSUBRQ' USING HCONN, HSUB, ACTION, SUBRQOPTS, COMPCODE, REASON.
```

Declare os parâmetros conforme a seguir:

```
** Connection handle
01 HCONN PIC S9(9) BINARY.
** Subscription handle
01 HSUB PIC S9(9) BINARY.
** Action requested by MQSUBRQ
01 ACTION PIC S9(9) BINARY.
** Options that control the action of MQSUBRQ
01 SUBRQOPTS.
COPY CMQSROV.
** Completion code
01 COMPCODE PIC S9(9) BINARY.
** Reason code qualifying COMPCODE
01 REASON PIC S9(9) BINARY.
```

## Chamada PL/I

```
call MQSUBRQ (Hconn, Hsub, Action, SubRqOpts, CompCode, Reason)
```

Declare os parâmetros conforme a seguir:

```
dcl Hconn fixed bin(31); /* Connection handle */
dcl Hsub fixed bin(31); /* Subscription handle */
dcl Action fixed bin(31); /* Action requested by MQSUBRQ */
dcl SubRqOpts like MQSRO; /* Options that control the action of MQSUBRQ */
dcl CompCode fixed bin(31); /* Completion code */
dcl Reason fixed bin(31); /* Reason code qualifying CompCode */
```

## Chamada do Assembler de Alto Nível

```
CALL MQSUBRQ,(HCONN, HSUB, ACTION, SUBRQOPTS,COMPCODE,REASON)
```

Declare os parâmetros conforme a seguir:

```
HCONN DS F Connection handle
HSUB DS F Subscription handle
ACTION DS F Action requested by MQSUBRQ
SUBRQOPTS CMQSROA , Options that control the action of MQSUBRQ
COMPCODE DS F Completion code
REASON DS F Reason code qualifying COMPCODE
```

## atributos de objetos

Esta coleção de tópicos lista apenas os objetos IBM MQ que podem ser o sujeito de uma chamada de função MQINQ e fornece detalhes dos atributos que podem ser consultados e os seletores a serem usados...

### Atributos do gerenciador de filas

Alguns atributos do gerenciador de filas são corrigidos para implementações específicas; outros podem ser alterados usando o comando MQSC ALTER QMGR.

Os atributos também podem ser exibidos usando o comando DISPLAY QMGR A maioria dos atributos do gerenciador de filas podem ser consultados abrindo um objeto MQOT\_Q\_MGR especial e usando a chamada MQINQ com o identificador retornado.

A tabela a seguir resume os atributos específicos do gerenciador de filas. Os atributos são descritos em ordem alfabética

**Nota:** Os nomes dos atributos mostrados nesta seção são nomes descritivos usados com a chamada MQINQ; os nomes são os mesmos que para os comandos PCF Quando comandos MQSC são usados para definir, alterar ou exibir atributos, nomes abreviados alternativos são usados; consulte [Comandos MQSC](#) para obter mais informações.

Atributo	Descrição
<a href="#">AccountingConnOverride</a>	Substituir configurações de contabilidade.
<a href="#">AccountingInterval</a>	Com que frequência gravar registros de contabilidade intermediária
<a href="#">ActivityConnOverride</a>	Substituir configurações de atividade.
<a href="#">ActivityTrace</a>	Controla a coleção do rastreo de atividade do aplicativo MQI IBM MQ .
<a href="#">AdoptNewMCACheck</a>	Elementos verificados para determinar se deve adotar novo MCA.
<a href="#">AdoptNewMCAType</a>	Se deve reiniciar automaticamente uma instância órfã de um MCA de um tipo de canal específico.
<a href="#">AlterationDate</a>	Data em que a definição foi alterada pela última vez
<a href="#">AlterationTime</a>	Horário em que a definição foi alterada pela última vez
<a href="#">AuthorityEvent</a>	Controla se eventos de autorização (Não Autorizado) são gerados
<a href="#">BridgeEvent</a>	Atributo de controle para eventos de ponte
<a href="#">ChannelAutoDef</a>	Controla se a definição de canal automático é permitida.
<a href="#">ChannelAutoDefEvent</a>	Controla se os eventos de definição automática do canal são gerados
<a href="#">ChannelAutoDefExit</a>	Nome da saída de usuário para definição de canal automática
<a href="#">ChannelEvent</a>	Atributo de controle para eventos do canal.
<a href="#">ChannelInitiatorControl</a>	Atributo de controle para inicializador de canais
<a href="#">ChannelMonitoring</a>	Dados de monitoramento on-line para canais
<a href="#">ChannelStatistics</a>	Controla a coleta de dados de estatísticas para canais
<a href="#">ChinitAdapters</a>	Número de subtarefas do adaptador para processar chamadas IBM MQ .
<a href="#">ChinitDispatchers</a>	Número de dispatchers a serem usados para o inicializador de canal.
	Reservado para uso IBM .
<a href="#">ChinitTraceAutoStart</a>	Se o rastreo do inicializador de canal deve ser iniciado automaticamente.
<a href="#">ChinitTraceTableSize</a>	Tamanho do espaço para dados de rastreo do inicializador de canais
<a href="#">ClusterSenderMonitoringDefault</a>	Padrão de dados de monitoramento on-line para canais do emissor de clusters
<a href="#">ClusterSenderStatistics</a>	Controla a coleção de informações de monitoramento de estatísticas para os canais do emissor de clusters
<a href="#">ClusterWorkloadData</a>	Dados do usuário para saída de carga de trabalho do cluster
<a href="#">ClusterWorkloadExit</a>	Nome da saída de usuário para gerenciamento de carga de trabalho do cluster
<a href="#">ClusterWorkloadLength</a>	Comprimento máximo de dados da mensagem transmitidos à saída de carga de trabalho do cluster

Tabela 557. Atributos do gerenciador de filas (continuação)


Atributo	Descrição
<a href="#">CLWLMRUChannels</a>	Número de canais usados mais recentemente para balanceamento de carga de trabalho do cluster
<a href="#">CLWLUseQ</a>	A carga de trabalho do cluster usa a fila remota
<a href="#">CodedCharSetId</a>	Identificador do conjunto de caracteres codificados
<a href="#">CommandEvent</a>	Atributo de controle para eventos de comando..
<a href="#">atributoCommandInputQName</a>	Nome da fila de entrada do comando
<a href="#">CommandLevel</a>	Nível de comando
<a href="#">atributo Controle doCommandServer</a>	Atributo de controle para o servidor de comando
<a href="#">Atributo Evento de Configuração</a>	Atributo de controle para eventos de configuração..
<a href="#">DeadLetterQName</a>	Nome da fila de mensagens não entregues
<a href="#">DefClusterXmitQueueTipo</a>	Tipo de fila de transmissão do cluster padrão
<a href="#">DefXmitQName</a>	Nome da Fila de Transmissão Padrão
<a href="#">DistLists</a>	Suporte à lista de distribuição
<a href="#">DNSGroup</a>	Nome do grupo para o listener TCP ao usar o suporte do Workload Manager Dynamic Domain Name Services.
<a href="#">DNSWLM</a>	Se o listener TCP é registrado com o Workload Manager for Dynamic Domain Name Services.
<a href="#">ExpiryInterval</a>	Intervalo entre varreduras para mensagens expiradas
<a href="#">IGQPutAuthority</a>	Autoridade put de enfileiramento intragrupo
<a href="#">IGQUserId</a>	Identificador de usuário de enfileiramento intragrupo
<a href="#">InhibitEvent</a>	Controla se os eventos de inibição (Inhibit Get e Inhibit Put) são gerados
 <a href="#">InitialKey 1</a>	A chave inicial para o sistema de proteção de senha.
<a href="#">IPAddressVersion</a>	Versão do endereço Internet Protocol
<a href="#">IntraGroupqueuing</a>	Suporte ao enfileiramento intragrupo
<a href="#">ListenerTimer</a>	Intervalo de tempo entre as tentativas de reiniciar o listener após falha de APPC ou TCP/IP.
<a href="#">LocalEvent</a>	Controla se eventos de erro locais são gerados
<a href="#">LoggerEvent</a>	Controla se eventos do criador de logs são gerados
<a href="#">LUGroupName</a>	Nome da LU genérica para listener da LU 6.2 que manipula transmissões de entrada para o grupo de filas compartilhadas.
<a href="#">LUName</a>	Nome da LU a ser usado para transmissões da LU de saída 6.2
<a href="#">LU62ARMSuffix</a>	Sufixo de SYS1.PARMLIB membro APPCPMxx, que designa LUADD para este inicializador de canais.
<a href="#">LU62Channels</a>	Número máximo de canais atuais ou clientes conectados que usam a LU 6.2
<a href="#">MaxActiveChannels</a>	O número máximo de canais que podem estar ativos a qualquer momento
<a href="#">MaxChannels</a>	O número máximo de canais atuais
<a href="#">MaxHandles</a>	Número máximo de identificadores
<a href="#">MaxMsgLength</a>	Tamanho máximo da mensagem em bytes
<a href="#">Atributo MaxPriority</a>	Prioridade máxima
<a href="#">MaxPropertiesLength</a>	Comprimento máximo de dados de propriedade em bytes
<a href="#">MaxUncommittedMsgs</a>	Número máximo de mensagens não confirmadas em uma unidade de trabalho
<a href="#">MQIAccounting</a>	Controla a coleta de informações de contabilidade para dados de MQI.
<a href="#">MQIStatistics</a>	Controla a coleta de informações de monitoramento de estatísticas para o gerenciador de filas
<a href="#">MsgMarkBrowseInterval</a>	Intervalo após o qual o gerenciador de filas pode remover a marca de mensagens procuradas.
<a href="#">OutboundPortMin</a>	Com <i>OutboundPortMin</i> , define o intervalo de números de porta a ser usado ao ligar os canais de saída
<a href="#">OutboundPortMax</a>	Com <i>OutboundPortMax</i> , define o intervalo de números de porta a ser usado ao ligar os canais de saída
<a href="#">PerformanceEvent</a>	Controla se eventos relacionados ao desempenho são gerados



Tabela 557. Atributos do gerenciador de filas (continuação)

Atributo	Descrição
<a href="#">Plataforma</a>	Plataforma na qual o gerenciador de filas está em execução
<a href="#">PubSubNPInputMsg</a>	Se descartar (ou manter) uma mensagem de entrada não entregue
<a href="#">PubSubNPResponse</a>	Controla o comportamento de não entregue
<a href="#">PubSubMaxMsgRetryCount</a>	O número de novas tentativas ao processar (sob ponto de sincronização) uma mensagem de comando com falha
<a href="#">PubSubSyncPoint</a>	Se apenas mensagens persistentes (ou todas as mensagens) devem ser processadas sob o ponto de sincronização
<a href="#">PubSubMode</a>	Se a interface de publicação / assinatura enfileirada está em execução.
<a href="#">QMGrDesc</a>	Descrição do gerenciador de filas
<a href="#">QMGrIdentifier</a>	Identificador exclusivo gerado internamente do gerenciador de filas
<a href="#">QMGrName</a>	Nome do gerenciador de filas
<a href="#">QSGName</a>	Nome do grupo de filas compartilhadas
<a href="#">QueueAccounting</a>	Controla a coleta de informações de contabilidade para filas.
<a href="#">QueueMonitoring</a>	Dados de monitoramento on-line para filas
<a href="#">QueueStatistics</a>	Controla a coleta de dados de estatísticas para filas
<a href="#">ReceiveTimeout</a>	Quanto tempo o canal TCP/IP aguarda dados antes de retornar ao estado inativo.
<a href="#">ReceiveTimeoutMin</a>	Qualificador para <i>ReceiveTimeout</i> ..
<a href="#">ReceiveTimeoutType</a>	Tempo mínimo que o canal TCP/IP aguarda dados antes de retornar ao estado inativo.
<a href="#">RemoteEvent</a>	Controla se eventos de erro remotos são gerados
<a href="#">RepositoryName</a>	Nome do cluster para o qual este gerenciador de filas fornece serviços de repositório
<a href="#">RepositoryNamelist</a>	Nome do objeto da lista de nomes contendo nomes de clusters para os quais este gerenciador de filas fornece serviços do repositório..
<a href="#">ScyCase</a>	Caso de perfis de segurança
<a href="#">SharedQMGr</a>	Nome do gerenciador de filas compartilhadas
<a href="#">"SPLCAP" na página 861</a>	IBM MQ Proteção de segurança de Mensagem Avançada para um gerenciador de filas ativado ou desativado
<a href="#">SSLCRLNamelist 1</a>	Nome do objeto da lista de nomes que contém nomes de objetos de informações sobre autenticação
<a href="#">SSLCryptoHardware 1</a>	Sequência de configuração de hardware criptográfico
<a href="#">SSLEvent</a>	Atributo de controle para eventos de TLS
<a href="#">SSLFIPSRequired</a>	Use apenas algoritmos certificados pelo FIPS para criptografia
<a href="#">SSLKeyRepository 1</a>	Local do repositório de chaves do TLS
 <a href="#">SSLKeyRepositorySenha 1</a>	A senha do repositório de chaves TLS.
<a href="#">SSLKeyResetCount</a>	Contagem de reconfiguração de chave do TLS
<a href="#">SSLTasks 1</a>	Número de subtarefas do servidor para processar chamadas TLS.
<a href="#">StatisticsInterval</a>	Com que frequência gravar os dados de monitoramento de estatísticas
<a href="#">StartStopEvent</a>	Controla se eventos de início e parada são gerados
<a href="#">SyncPoint</a>	Disponibilidade do ponto de sincronização
<a href="#">TCPChannels</a>	O número máximo de canais atuais ou clientes conectados que usam TCP/IP.
<a href="#">TCPKeepAlive</a>	Se deve usar TCP KEEPALIVE para verificar outra extremidade da conexão.
<a href="#">TCPName</a>	O nome do sistema TCP/IP que você está usando
<a href="#">TCPStackType</a>	Como o inicializador de canais pode utilizar endereços TCP/IP.
<a href="#">atributo Gravação doTraceRoute</a>	Controla a gravação de informações de rastreamento de rotas..
<a href="#">TriggerInterval</a>	Acionador-intervalo de mensagem
<a href="#">Versão</a>	Versão
<a href="#">XrCapability</a>	Especifica se os comandos de Telemetria são suportados

Tabela 557. Atributos do gerenciador de filas (continuação)

Atributo	Descrição
<b>Notas:</b>	
1. Este atributo não pode ser consultado usando a chamada MQINQ e não está descrito nesta seção Consulte <a href="#">Change Queue Manager</a> para obter detalhes deste atributo.	

### Tarefas relacionadas

[Especificando que Apenas CipherSpecs Certificados por FIPS São Usados no Tempo de Execução no Cliente de MQI](#)

### Referências relacionadas

[Federal Information Processing Standards \(FIPS\) para AIX, Linux, and Windows](#)

## AccountingConnSubstituição (MQLONG)

Isso permite que os aplicativos substituam a configuração dos valores ACCTMQI e ACCTQDATA no atributo Qmgr.

O valor é um dos seguintes:

### MQMON\_DISABLED

Os aplicativos não podem substituir a configuração dos atributos ACCTMQI e ACCTQ Qmgr usando o campo Opções na estrutura MQCNO na chamada MQCONNX Esse é o valor-padrão.

### MQMON\_ENABLED

Os aplicativos podem substituir os atributos ACCTQ e ACCTMQI Qmgr usando o campo Opções na estrutura MQCNO

As mudanças nesse valor são efetivas apenas para conexões com o gerenciador de filas após a mudança no atributo.

Esse atributo é suportado apenas nas seguintes plataformas:

- ▶ **IBM i** IBM i
- ▶ **Linux** ▶ **AIX** AIX and Linux
- ▶ **Windows** Windows

Para determinar o valor desse atributo, use o seletor MQIA\_ACCOUNTING\_CONN\_OVERRIDE com a chamada MQINQ..

## AccountingInterval (MQLONG)

Isso especifica quanto tempo antes de registros de contabilidade intermediários serem gravados (em segundos).

O valor é um número inteiro no intervalo de 0 a 604800, com um valor padrão de 1800 (30 minutos). Especifique 0 para desativar registros intermediários.

Esse atributo é suportado apenas nas seguintes plataformas:

- ▶ **IBM i** IBM i
- ▶ **Linux** ▶ **AIX** AIX and Linux
- ▶ **Linux** Linux
- ▶ **Windows** Windows

Para determinar o valor desse atributo, use o seletor MQIA\_ACCOUNTING\_INTERVAL com a chamada MQINQ.

## Substituição de ActivityConn(MQLONG)

Isso permite que aplicativos substituam a configuração do valor ACTVTRC no atributo do gerenciador de filas.

O valor é um dos seguintes:

**MQMON\_DISABLED**

Os aplicativos não podem substituir a configuração do atributo do gerenciador de filas ACTVTRC usando o campo Opções na estrutura MQNC na chamada MQCONN. Esse é o valor-padrão.

**MQMON\_ENABLED**

Os aplicativos podem substituir o atributo do gerenciador de filas ACTVTRC usando o campo Opções na estrutura MQCNO.

As mudanças nesse valor são efetivas apenas para conexões com o gerenciador de filas após a mudança no atributo.

Esse atributo é suportado apenas no [Multiplataformas](#)

Para determinar o valor desse atributo, use o seletor MQIA\_ACTIVITY\_CONN\_OVERRIDE com a chamada MQINQ

**ActivityTrace (MQLONG)**

Isso controla a coleção do rastreo de atividade do aplicativo MQI IBM MQ .

O valor é um dos seguintes:

**MQMON\_ON**

Colete o rastreo de atividade do aplicativo MQI IBM MQ .

**MQMON\_OFF**

Não colete o rastreo de atividade do aplicativo MQI IBM MQ . Esse é o valor-padrão.

Se você configurar o atributo do gerenciador de filas ACTVCON0 como ENABLED, esse valor poderá ser substituído para conexões individuais usando o campo Opções na estrutura MQCNO.

As mudanças nesse valor são efetivas apenas para conexões com o gerenciador de filas após a mudança no atributo.

Esse atributo é suportado apenas no [Multiplataformas](#)

Para determinar o valor desse atributo, use o seletor MQIA\_ACTIVITY\_TRACE com a chamada MQINQ

**AdoptNewMCACheck (MQLONG)**

Isso define os elementos a serem verificados para determinar se deve adotar um MCA quando um novo canal de entrada for detectado que tenha o mesmo nome que um MCA que já está ativo

O valor é um dos seguintes:

**MQADOPT\_CHECK\_Q\_MGR\_NAME**

Verifique o nome do gerenciador de filas.

**MQADOPT\_CHECK\_NET\_ADDR**

Verifique o endereço da rede.


**MQADOPT\_CHECK\_ALL**

Verifique o nome do gerenciador de filas e o endereço de rede Se possível, execute essa verificação para proteger seus canais de serem desligados, inadvertidamente ou maliciosamente. Esse é o valor-padrão.

**MQADOPT\_CHECK\_NONE**

Não verifique nenhum elemento

As mudanças nesse atributo entram em vigor na próxima vez que um canal tentar adotar um canal.

 Esse atributo é suportado apenas no z/OS

Para determinar o valor desse atributo, use o seletor MQIA\_ADOPTNEWMCA\_CHECK com a chamada MQINQ

### ***AdoptNewMCAType (MQLONG)***

Isso especifica se deve reiniciar automaticamente uma instância órfã de um MCA de um tipo de canal específico quando uma nova solicitação de canal de entrada correspondente ao atributo MCACheck AdoptNewfor detectado

Ele é um dos seguintes valores:

#### **MQADOPT\_TYPE\_NO**

Não é necessário adotar instâncias de canal órfãs. Esse é o valor-padrão.

#### **MQADOPT\_TYPE\_ALL**

Adote todos os tipos de canal.

Esse atributo é suportado apenas no z/OS

Para determinar o valor desse atributo, use o seletor MQIA\_ADOPTNEWMCA\_TYPE com a chamada MQINQ..

### ***AlterationDate (MQCHAR12)***

Esta é a data em que a definição foi mudada pela última vez O formato da data é YYYY-MM-DD, preenchido com dois espaços em branco finais para tornar o comprimento de 12 bytes.

Para determinar o valor desse atributo, use o seletor MQCA\_ALTERATION\_DATE com a chamadas MQINQ. O comprimento desse atributo é fornecido por MQ\_DATE\_LENGTH

### ***AlterationTime (MQCHAR8)***

Este é o horário em que a definição foi alterada pela última vez O formato do horário é HH.MM.SS

Para determinar o valor desse atributo, use o seletor MQCA\_ALTERATION\_TIME com a chamada MQINQ.. O comprimento desse atributo é fornecido por MQ\_TIME\_LENGTH

### ***AuthorityEvent (MQLONG)***

Isso controla se os eventos de autorização (Não Autorizado) são gerados Ele é um dos seguintes valores:

#### **MQEVR\_DISABLED**

Relatório de eventos desativado.

#### **MQEVR\_ENABLED**

Relatório de eventos ativado

Para obter mais informações sobre os eventos, consulte [Monitoramento de eventos](#)

Para determinar o valor desse atributo, use o seletor MQIA\_AUTHORITY\_EVENT com a chamada MQINQ..

### ***BridgeEvent (MQLONG)***

Isso especifica se eventos de ponte IMS são gerados.

O valor é um dos seguintes:

#### **MQEVR\_ENABLED**

Gere eventos de ponte IMS , conforme a seguir:

MQRC\_BRIDGE\_STARTED

MQRC\_BRIDGE\_STOPPED

#### **MQEVR\_DISABLED**

Não gere eventos de ponte IMS ; este é o valor padrão.

Esse atributo é suportado apenas no z/OS

Para determinar o valor desse atributo, use o seletor MQIA\_BRIDGE\_EVENT com a chamada MQINQ..

### ***ChannelAutoDef (MQLONG)***

Este atributo controla a definição automática de canais do tipo MQCHT\_RECEIVER e MQCHT\_SVRCONN. A definição automática de canais MQCHT\_CLUSSDR é sempre ativada. O valor é um dos seguintes:

**MQCHAD\_DISABLED**

Definição automática de canal desativada.

**MQCHAD\_ENABLED**

Definição automática de canal ativada.

 Esse atributo é suportado apenas no [Multiplataformas](#)

Para determinar o valor desse atributo, use o seletor MQIA\_CHANNEL\_AUTO\_DEF com a chamada MQINQ..

***ChannelAutoDefEvent (MQLONG)***

Isso controla se eventos de definição automática de canal são gerados. Ele se aplica a canais do tipo MQCHT\_RECEIVER, MQCHT\_SVRCONN e MQCHT\_CLUSSDR.. O valor é um dos seguintes:

**MQEVR\_DISABLED**

Relatório de eventos desativado.

**MQEVR\_ENABLED**

Relatório de eventos ativado

Para obter mais informações sobre os eventos, consulte [Monitoramento de eventos](#)

 Esse atributo é suportado apenas no [Multiplataformas](#)


Para determinar o valor desse atributo, use o seletor MQIA\_CHANNEL\_AUTO\_DEF\_EVENT com a chamada MQINQ..

***ChannelAutoDefExit (MQCHARn)***

Este é o nome da saída de usuário para definição de canal automática. Se esse nome não estiver em branco e *ChannelAutoDef* tiver o valor MQCHAD\_ENABLED, a saída será chamada toda vez que o gerenciador de filas estiver prestes a criar uma definição de canal. Isso se aplica aos canais do tipo MQCHT\_RECEIVER, MQCHT\_SVRCONN, e MQCHT\_CLUSSDR. A saída pode, então, executar um dos seguintes procedimentos:

- Crie a definição de canal sem alteração..
- Modifique os atributos da definição de canal criada.
- Suprimir a criação do canal inteiramente.

**Nota:** O comprimento e o valor desse atributo são específicos do ambiente.. Consulte a introdução à estrutura MQCD em “MQCD-Definição de canal” na [página 1532](#) para obter detalhes do valor desse atributo em vários ambientes.

 No z/OS, esse atributo aplica-se apenas aos canais do emissor de clusters e do receptor de clusters

Para determinar o valor desse atributo, use o seletor MQCA\_CHANNEL\_AUTO\_DEF\_EXIT com a chamada MQINQ. O comprimento desse atributo é fornecido por MQ\_EXIT\_NAME\_LENGTH

***ChannelEvent (MQLONG)***

Isso especifica se eventos do canal são gerados.

Ele é um dos seguintes valores:

**MQEVR\_EXCEPTION**

Gere apenas os seguintes eventos do canal:

- MQRC\_CHANNEL\_ACTIVATED
- MQRC\_CHANNEL\_CONV\_ERROR

- MQRC\_CHANNEL\_NOT\_ACTIVATED
- MQRC\_CHANNEL\_STOPPED com os seguintes ReasonQualifiers:
  - MQRQ\_CHANNEL\_STOPPED\_ERROR
  - MQRQ\_CHANNEL\_STOPPED\_RETRY
  - MQRQ\_CHANNEL\_STOPPED\_DISABLED
- MQRC\_CHANNEL\_STOPPED\_BY\_USER

#### **MQEVR\_ENABLED**

Gere todos os eventos do canal Ou seja, além daqueles gerados por EXCEPTION, gere os seguintes eventos do canal:

- MQRC\_CHANNEL\_STARTED
- MQRC\_CHANNEL\_STOPPED com o seguinte ReasonQualifier:
  - MQRQ\_CHANNEL\_STOPPED\_OK

#### **MQEVR\_DISABLED**

Não gerar eventos do canal; este é o valor padrão.

Para determinar o valor desse atributo, use o seletor MQIA\_CHANNEL\_EVENT com a chamada MQINQ..

#### ***Controle do ChannelInitiator(MQLONG).***

Especifica se o inicializador de canais deve ser iniciado quando o gerenciador de filas for iniciado.

Ele é um dos seguintes valores:

#### **MQSVC\_CONTROL\_MANUAL**

O inicializador de canais não será iniciado automaticamente.

#### **MQSVC\_CONTROL\_Q\_MGR**

O inicializador de canais deve ser iniciado automaticamente quando o gerenciador de filas for iniciado.

Para determinar o valor desse atributo, use o seletor MQIA\_CHINIT\_CONTROL com a chamada MQINQ..

#### ***ChannelMonitoring (MQLONG)***

Este atributo especifica os dados de monitoramento on-line para canais

O valor é um dos seguintes:

#### **MQMON\_NONE**

Desative a coleta de dados para monitoramento do canal para todos os canais, independentemente da configuração do atributo do canal MONCHL. Esse é o valor-padrão.

#### **MQMON\_OFF**

Desative a coleta de dados de monitoramento para canais que especificam QMGR no atributo do canal MONCHL.

#### **MQMON\_LOW**


Ative a coleta de dados de monitoramento com uma razão baixa de coleta de dados para canais que especificam QMGR no atributo do canal MONCHL..

#### **MQMON\_MEDIUM**

Ative a coleta de dados de monitoramento com uma proporção moderada de coleta de dados para canais que especificam QMGR no atributo do canal MONCHL.

#### **MQMON\_HIGH**

Ative a coleta de dados de monitoramento com uma alta proporção de coleta de dados para canais especificando QMGR no atributo do canal MONCHL.

 Em sistemas z/OS , ativar esse parâmetro simplesmente ativa a coleta de dados estatísticos, independentemente do valor selecionado. Especificar LOW, MEDIUM, ou HIGH não faz diferença para seus resultados.

Para determinar o valor desse atributo, use o seletor MQIA\_MONITORING\_CHANNEL com a chamadas MQINQ.

### ***ChannelStatistics (MQLONG)***

Isso controla a coleção de dados de estatísticas para canais

O valor é um dos seguintes:

#### **MQMON\_NONE**

Desative a coleção de dados para estatísticas do canal para todos os canais, independentemente da configuração do atributo do canal STATCHL. Esse é o valor-padrão.

#### **MQMON\_OFF**

Desative a coleta de dados estatísticos para canais que especificam QMGR no atributo do canal STATCHL.

#### **MQMON\_LOW**

Ative a coleta de dados estatísticos com uma baixa proporção de coleta de dados para canais especificando QMGR no atributo do canal STATCHL.


#### **MQMON\_MEDIUM**

Ative a coleta de dados estatísticos com uma taxa moderada de coleta de dados para canais especificando QMGR no atributo do canal STATCHL.

#### **MQMON\_HIGH**

Ative a coleta de dados de estatísticas com uma alta proporção de coleta de dados para canais que especificam QMGR no atributo do canal STATCHL

Para a maioria dos sistemas, recomenda-se usar MEDIUM. No entanto, para um canal que processa um alto volume de mensagens por segundo, talvez você queira reduzir o nível de amostragem selecionando LOW. Além disso, para um canal que processa apenas algumas mensagens e para o qual as informações mais atuais são importantes, você pode selecionar HIGH.

 Em sistemas z/OS, ativar esse parâmetro simplesmente ativa a coleta de dados estatísticos, independentemente do valor selecionado. Especificar LOW, MEDIUM, ou HIGH não faz diferença para seus resultados. Esse parâmetro deve ser ativado para coletar registros de contabilidade de canal.

Para determinar o valor desse atributo, use o seletor MQIA\_STATISTICS com a chamada MQINQ.

### ***ChinitAdapters (MQLONG)***

Esse é o número de subtarefas do adaptador a serem usadas para processar chamadas de IBM MQ. O valor deve ser 0-9999, com um valor padrão de 8.

A proporção de adaptadores para dispatchers (o atributo ChinitDispatchers) deve ser de 8 a 5. No entanto, se você tiver apenas alguns canais, não será necessário diminuir o valor desse parâmetro do valor padrão. É possível usar os seguintes valores: para um sistema de teste, 8 (padrão); para um sistema de produção, 20. Idealmente, você deve ter 20 adaptadores, que fornecem maior paralelismo de chamadas IBM MQ. Isso é importante para mensagens persistentes. Menos adaptadores podem ser melhores para mensagens não persistentes

Esse atributo é suportado apenas no z/OS

Para determinar o valor desse atributo, use o seletor MQIA\_CHINIT\_ADAPTERS com a chamada MQINQ.

### ***ChinitDispatchers (MQLONG)***

Esse é o número de dispatchers a serem usados para o inicializador de canais.. O valor deve ser de 0 a 9999, com um valor padrão de 5.

Como uma diretriz, permita um dispatcher para 50 canais atuais. No entanto, se você tiver apenas alguns canais, não será necessário diminuir o valor desse atributo do valor padrão. Se você estiver usando TCP/IP, o maior número de dispatchers que são usados para canais TCP/IP será 100, mesmo se você especificar um valor maior aqui. Você pode usar as seguintes configurações: sistemas de teste, 5 (o

padrão); sistemas de produção, 20 (você precisa de 20 dispatchers para manipular até 1000 canais ativos).

Esse atributo é suportado apenas no z/OS

Para determinar o valor desse atributo, use o seletor MQIA\_CHINIT\_DISPATCHERS com a chamada MQINQ..

### ***ChinitTraceAutoStart (MQLONG).***

Isso especifica se o rastreamento do inicializador de canais deve ser iniciado automaticamente.

O valor é um dos seguintes:

#### **MQTRAXSTR\_YES**

Inicie o rastreamento do inicializador de canais automaticamente. Esse é o valor-padrão.

#### **MQTRAXSTR\_NO**

Não inicie o rastreamento do inicializador de canais automaticamente.

Esse atributo é suportado apenas no z/OS

Para determinar o valor desse atributo, use o seletor MQIA\_CHINIT\_TRACE\_AUTO\_START com a chamada MQINQ..

### ***ChinitTraceTableSize (MQLONG)***

Esse é o tamanho do espaço para dados de rastreamento do inicializador de canais (em MB)..

O valor deve estar no intervalo de 0 a 2048, com um valor padrão 2.

**Nota:** Sempre que você usar grandes espaços para dados do z/OS , assegure-se de que tenha armazenamento auxiliar suficiente em seu sistema para suportar qualquer atividade de paginação relacionada do z/OS Talvez você precise aumentar o tamanho dos conjuntos de dados SYS1.DUMP.

Esse atributo é suportado apenas no z/OS

Para determinar o valor desse atributo, use o seletor MQIA\_CHINIT\_TRACE\_TABLE\_SIZE com a chamada MQINQ..

### ***ClusterSenderMonitoringDefault (MQLONG)***

Isso especifica o valor a ser substituído pelo atributo ChannelMonitoring de canais do emissor de clusters definidos automaticamente.

O valor é um dos seguintes:

#### **MQMON\_Q\_MGR**

A coleção de dados de monitoramento on-line é herdada da configuração do atributo **ChannelMonitoring** do gerenciador de filas Esse é o valor-padrão.

#### **MQMON\_OFF**

O monitoramento para o canal está desativado.

#### **MQMON\_LOW**

A menos que *ChannelMonitoring* seja MQMON\_NONE, o monitoramento é ativado com uma taxa baixa de coleta de dados com um efeito mínimo sobre o desempenho do sistema Os dados coletados provavelmente não serão os mais atuais.

#### **MQMON\_MEDIUM**

A menos que *ChannelMonitoring* seja MQMON\_NONE, o monitoramento é ativado com uma taxa moderada de coleta de dados com efeito limitado sobre o desempenho do sistema

#### **MQMON\_HIGH**

A menos que *ChannelMonitoring* seja MQMON\_NONE, o monitoramento é ativado com uma alta taxa de coleta de dados com um provável efeito sobre o desempenho do sistema Os dados coletados são os mais atuais disponíveis.



Para determinar o valor desse atributo, use o seletor MQIA\_MONITORING\_AUTO\_CLUSSDR com a chamada MQINQ..

### ***ClusterSenderEstatísticas (MQLONG)***

Como os canais do emissor de clusters podem ser definidos automaticamente a partir da definição de CLUSRCVR no repositório, não é possível alterar a configuração do atributo STATCHL para esses canais do emissor de clusters definidos automaticamente usando o canal ALTER.. Para esses canais, a decisão de coletar dados de monitoramento on-line é baseada na configuração desse atributo do gerenciador de filas.

O valor é um dos seguintes:

#### **MQMON\_Q\_MGR**

A coleta de dados estatísticos para canais do emissor de clusters definidos automaticamente é baseada no valor do atributo STATCHL do gerenciador de filas. Esse é o valor-padrão.

#### **MQMON\_OFF**

Desative a coleta de dados de estatísticas para canais do emissor de cluster definidos automaticamente

#### **MQMON\_LOW**

Ative a coleta de dados estatísticos para canais do emissor de clusters definidos automaticamente, com uma baixa proporção de coleta de dados


#### **MQMON\_MEDIUM**

Ative a coleta de dados estatísticos para canais do emissor de clusters definidos automaticamente com uma proporção moderada de coleta de dados

#### **MQMON\_HIGH**

Ative a coleta de dados estatísticos para canais do emissor de clusters auto-definidos com uma alta proporção de coleta de dados

Para a maioria dos sistemas, recomendamos o MEDIUM. No entanto, para um canal emissor de cluster definido automaticamente que processa um alto volume de mensagens a cada segundo, talvez você queira reduzir o nível de amostragem selecionando LOW. Além disso, para um canal que processa apenas algumas mensagens e para o qual as informações mais atuais são importantes, você pode selecionar HIGH.

 Em sistemas z/OS , ativar esse parâmetro simplesmente ativa a coleta de dados estatísticos, independentemente do valor selecionado. Especificar LOW, MEDIUM, ou HIGH não faz diferença para seus resultados. Esse parâmetro deve ser ativado para coletar registros de contabilidade de canal.

Para determinar o valor desse atributo, use o seletor MQIA\_STATISTICS\_AUTO\_CLUSSDR com a chamada MQINQ.

### ***ClusterWorkloadDados (MQCHAR32)***

Essa é uma sequência de caracteres de 32 bytes definida pelo usuário que é transmitida para a saída de carga de trabalho do cluster quando ela é chamada. Se não houver dados a serem transmitidos para a saída, a sequência estará em branco.

Para determinar o valor desse atributo, use o seletor MQCA\_CLUSTER\_WORKLOAD\_DATA com a chamada MQINQ.

### ***Saída do ClusterWorkload(MQCHARn)***

Este é o nome da saída de usuário para o gerenciamento de carga de trabalho do cluster. Se esse nome não estiver em branco, a saída será chamada toda vez que uma mensagem for colocada em uma fila de clusters ou movida de uma fila do emissor de clusters para outra. A saída pode então aceitar a instância da fila selecionada pelo gerenciador de filas como o destino para a mensagem ou selecionar outra instância da fila.

**Nota:** O comprimento e o valor desse atributo são específicos do ambiente..

Para determinar o valor desse atributo, use o seletor MQCA\_CLUSTER\_WORKLOAD\_EXIT com a chamada MQINQ.. O comprimento desse atributo é fornecido por MQ\_EXIT\_NAME\_LENGTH

### ***Comprimento do ClusterWorkload(MQLONG)***

Este é o comprimento máximo de dados da mensagem que é transmitido para a saída de carga de trabalho do cluster O comprimento real dos dados passados para a saída é o mínimo dos seguintes:

- O comprimento da mensagem.
- O atributo **MaxMsgLength** do Gerenciador de Filas
- O atributo **ClusterWorkloadLength** ..

Para determinar o valor desse atributo, use o seletor MQIA\_CLUSTER\_WORKLOAD\_LENGTH com a chamada MQINQ

### ***CLWLMRUChannels (MQLONG)***

Isso especifica o número máximo de canais de cluster usados mais recentemente, a serem considerados para uso pelo algoritmo de opção de carga de trabalho do cluster

Este é um valor no intervalo de 1 a 999999999

Para determinar o valor desse atributo, use o seletor MQIA\_CLWL\_MRU\_CHANNELS com a chamada MQINQ..

### ***CLWLUseQ (MQLONG)***

Isso especifica se deve usar filas remotas para a carga de trabalho do cluster.

O valor é um dos seguintes:

#### **MQCLWL\_USEQ\_ANY**

Use filas locais e remotas.

#### **MQCLWL\_USEQ\_LOCAL**

Não use filas remotas.. Esse é o valor-padrão.

Para determinar o valor desse atributo, use o seletor MQIA\_CLWL\_USEQ com a chamada MQINQ.

### ***CodedCharSetId (MQLONG)***

Isso define o conjunto de caracteres usado pelo gerenciador de filas para todos os campos de sequência de caracteres definidos no MQI, como os nomes dos objetos e a data e hora de criação da fila O conjunto de caracteres deve ser um que tenha caracteres de byte único para os caracteres válidos em nomes de objetos. Ele não se aplica aos dados do aplicativo transportados na mensagem O valor depende do ambiente:

- No z/OS, o valor é configurado a partir dos parâmetros do sistema quando o gerenciador de filas é iniciado; o valor padrão é 500..
- No Windows, o valor é CODEPAGE primário do usuário que está criando o gerenciador de filas.
- No IBM i, o valor é aquele configurado no ambiente quando o gerenciador de filas é criado pela primeira vez.
- No AIX and Linux, o valor é o CODESET padrão para o código de idioma do usuário que cria o gerenciador de filas.

Para determinar o valor desse atributo, use o seletor MQIA\_CODED\_CHAR\_SET\_ID com a chamada MQINQ..

### ***CommandEvent (MQLONG)***

Isso especifica se os eventos de comando são gerados, conforme a seguir:

**MQEVN\_DISABLED**

Não gere eventos de comando Esse é o padrão.

**MQEVN\_ENABLED**

Gere eventos de comando

**MQEVN\_NO\_DISPLAY**

Eventos de comando são gerados para todos os comandos bem-sucedidos diferentes de MQINQ.

Para determinar o valor desse atributo, use o seletor MQIA\_COMMAND\_EVENT com a chamada MQINQ..

**CommandInputQName (MQCHAR48)**

Este é o nome da fila de entrada de comandos definida no gerenciador de fila local Esta é uma fila para a qual os usuários podem enviar comandos, se autorizado a fazê-lo. O nome da fila depende do ambiente:

- No z/OS, o nome da fila é SYSTEM.COMMAND.INPUT; os comandos MQSC e PCF podem ser enviados para ele Consulte [Os comandos MQSC](#) para obter detalhes de comandos MQSC e [Definições dos Formatos de Comando Programáveis](#) para obter detalhes de comandos PCF
- Em todos os outros ambientes, o nome da fila é SYSTEM.ADMIN.COMMAND.QUEUEe apenas comandos PCF podem ser enviados para ele.. No entanto, um comando MQSC poderá ser enviado para essa fila se o comando MQSC estiver incluído em um comando PCF do tipo MQCMD\_ESCAPE.. Consulte [Escape](#) , para obter informações sobre o comando Escape

Para determinar o valor desse atributo, use o seletor MQCA\_COMMAND\_INPUT\_Q\_NAME com a chamada MQINQ. O comprimento desse atributo é fornecido por MQ\_Q\_NAME\_LENGTH

**CommandLevel (MQLONG)**

**Nota:** O suporte para o sistema operacional HP-UX para todos os componentes do IBM MQ , incluindo servidor e clientes, foi removido no IBM MQ 9.1.

Isso indica o nível de comandos de controle do sistema suportados pelo gerenciador de filas Este pode ser um dos valores a seguir:

**MQCMDL\_LEVEL\_800**

O nível 800 de comandos de controle do sistema.

Este valor é retornado pelas seguintes versões:

- IBM MQ for AIX 8.0
- IBM MQ for IBM i 8.0
- IBM MQ for Linux 8.0
- IBM MQ for Windows 8.0
- IBM MQ for z/OS 8.0

**MQCMDL\_LEVEL\_801**

Nível 801 de comandos de controle do sistema.

Este valor é retornado pelas seguintes versões:

- IBM MQ for AIX 8.0.0 Fix Pack 2
- IBM MQ for HP-UX 8.0.0 Fix Pack 2
- IBM MQ for IBM i 8.0.0 Fix Pack 2
- IBM MQ for Linux 8.0.0 Fix Pack 2

**MQCMDL\_LEVEL\_802**

Nível 802 de comandos de controle do sistema.

Este valor é retornado pelas seguintes versões:

- IBM MQ for AIX 8.0.0 Fix Pack 3
- IBM MQ for IBM i 8.0.0 Fix Pack 3

- IBM MQ for Linux 8.0.0 Fix Pack 3
- IBM MQ for Windows 8.0.0 Fix Pack 3

#### **MQCMDL\_LEVEL\_900**

O nível 900 dos comandos de controle do sistema

Este valor é retornado pelas seguintes versões:

- IBM MQ for AIX 9.0
- IBM MQ for IBM i 9.0
- IBM MQ for Linux 9.0
- IBM MQ for Windows 9.0
- IBM MQ for z/OS 9.0

#### **MQCMDL\_LEVEL\_901**

O nível 901 de comandos de controle do sistema.

Este valor é retornado pelas seguintes versões:

- IBM MQ for Linux 9.0.1
- IBM MQ for Windows 9.0.1
- IBM MQ for z/OS 9.0.1

#### **MQCMDL\_LEVEL\_902**

Nível 902 de comandos de controle do sistema.

Este valor é retornado pelas seguintes versões:

- IBM MQ for Linux 9.0.2
- IBM MQ for Windows 9.0.2
- IBM MQ for z/OS 9.0.2

#### **MQCMDL\_LEVEL\_903**

Nível 903 de comandos de controle do sistema..

Este valor é retornado pelas seguintes versões:

- IBM MQ for Linux 9.0.3
- IBM MQ for Windows 9.0.3
- IBM MQ for z/OS 9.0.3

#### **MQCMDL\_LEVEL\_904**

O nível 904 dos comandos de controle do sistema

Este valor é retornado pelas seguintes versões:

- IBM MQ for AIX 9.0.4
- IBM MQ for Linux 9.0.4
- IBM MQ for Windows 9.0.4
- IBM MQ for z/OS 9.0.4

#### **MQCMDL\_LEVEL\_905**

Nível 905 de comandos de controle do sistema.

Este valor é retornado pelas seguintes versões:

- IBM MQ for AIX 9.0.5
- IBM MQ for Linux 9.0.5
- IBM MQ for Windows 9.0.5
- IBM MQ for z/OS 9.0.5

**MQCMDL\_LEVEL\_910**

Nível 910 de comandos de controle do sistema.

Este valor é retornado pelas seguintes versões:

- IBM MQ for AIX 9.1
- IBM MQ for IBM i 9.1
- IBM MQ for Linux 9.1
- IBM MQ for Windows 9.1
- IBM MQ for z/OS 9.1

**MQCMDL\_LEVEL\_911**

Nível 911 de comandos de controle do sistema..

Este valor é retornado pelas seguintes versões:

- IBM MQ for AIX 9.1.1
- IBM MQ for Linux 9.1.1
- IBM MQ for Windows 9.1.1
- IBM MQ for z/OS 9.1.1

**MQCMDL\_LEVEL\_912**

Nível 912 de comandos de controle do sistema.

Este valor é retornado pelas seguintes versões:

- IBM MQ for AIX 9.1.2
- IBM MQ for Linux 9.1.2
- IBM MQ for Windows 9.1.2
- IBM MQ for z/OS 9.1.2

**MQCMDL\_LEVEL\_913**

Nível 913 de comandos de controle do sistema.

Este valor é retornado pelas seguintes versões:

- IBM MQ for AIX 9.1.3
- IBM MQ for Linux 9.1.3
- IBM MQ for Windows 9.1.3
- IBM MQ for z/OS 9.1.3

**MQCMDL\_LEVEL\_914**

Nível 914 de comandos de controle do sistema..

Este valor é retornado pelas seguintes versões:

- IBM MQ for AIX 9.1.4
- IBM MQ for Linux 9.1.4
- IBM MQ for Windows 9.1.4
- IBM MQ for z/OS 9.1.4

**MQCMDL\_LEVEL\_915**

Nível 915 de comandos de controle do sistema.

Este valor é retornado pelas seguintes versões:

- IBM MQ for AIX 9.1.5
- IBM MQ for Linux 9.1.5
- IBM MQ for Windows 9.1.5
- IBM MQ for z/OS 9.1.5

**MQCMDL\_LEVEL\_910**

Nível 910 de comandos de controle do sistema.

Este valor é retornado pelas seguintes versões:

- IBM MQ for AIX 9.1
- IBM MQ for IBM i 9.1
- IBM MQ for Linux 9.1
- IBM MQ for Windows 9.1
- IBM MQ for z/OS 9.1

**MQCMDL\_LEVEL\_920**

Nível 920 de comandos de controle do sistema.

Este valor é retornado pelas seguintes versões:

- IBM MQ for AIX 9.2
- IBM MQ for IBM i 9.2
- IBM MQ for Linux 9.2
- IBM MQ for Windows 9.2
- IBM MQ for z/OS 9.2

**MQCMDL\_LEVEL\_921**

Nível 921 de comandos de controle de sistema.

Este valor é retornado pelas seguintes versões:

- IBM MQ for AIX 9.2.1
- IBM MQ for Linux 9.2.1
- IBM MQ for Windows 9.2.1
- IBM MQ for z/OS 9.2.1

**MQCMDL\_LEVEL\_922**

O nível 922 dos comandos de controle do sistema

Este valor é retornado pelas seguintes versões:

- IBM MQ for AIX 9.2.2
- IBM MQ for Linux 9.2.2
- IBM MQ for Windows 9.2.2
- IBM MQ for z/OS 9.2.2

**MQCMDL\_LEVEL\_923**

O nível 923 dos comandos de controle do sistema

Este valor é retornado pelas seguintes versões:

- IBM MQ for AIX 9.2.3
- IBM MQ for Linux 9.2.3
- IBM MQ for Windows 9.2.3
- IBM MQ for z/OS 9.2.3

**MQCMDL\_LEVEL\_924**

Nível 924 de comandos de controle do sistema.

Este valor é retornado pelas seguintes versões:

- IBM MQ for AIX 9.2.4
- IBM MQ for Linux 9.2.4
- IBM MQ for Windows 9.2.4

- IBM MQ for z/OS 9.2.4

#### **MQCMDL\_LEVEL\_925**

Nível 925 de comandos de controle do sistema.

Este valor é retornado pelas seguintes versões:

- IBM MQ for AIX 9.2.5
- IBM MQ for Linux 9.2.5
- IBM MQ for Windows 9.2.5
- IBM MQ for z/OS 9.2.5

#### **MQCMDL\_LEVEL\_930**

Nível 930 de comandos de controle do sistema.

Este valor é retornado pelas seguintes versões:

- IBM MQ for AIX 9.3
- IBM MQ for IBM i 9.3
- IBM MQ for Linux 9.3
- IBM MQ for Windows 9.3
- IBM MQ for z/OS 9.3

#### **MQCMDL\_LEVEL\_931**

Nível 931 dos comandos de controle do sistema.

Este valor é retornado pelas seguintes versões:

- IBM MQ for AIX 9.3.1
- IBM MQ for Linux 9.3.1
- IBM MQ for Windows 9.3.1
- IBM MQ for z/OS 9.3.1

#### **MQCMDL\_LEVEL\_932**

Nível 932 de comandos de controle do sistema.

Este valor é retornado pelas seguintes versões:

- IBM MQ for AIX 9.3.2
- IBM MQ for Linux 9.3.2
- IBM MQ for Windows 9.3.2
- IBM MQ for z/OS 9.3.2

O conjunto de comandos de controle do sistema que corresponde a um determinado valor do atributo **CommandLevel** varia de acordo com o valor do atributo **Platform** ; ambos devem ser usados para decidir quais comandos de controle do sistema são suportados.

Para determinar o valor desse atributo, use o seletor MQIA\_COMMAND\_LEVEL com a chamada MQINQ

#### ***Controle do CommandServer(MQLONG).***

Especifica se o servidor de comando deve ser iniciado quando o gerenciador de filas for iniciado

O valor pode ser qualquer um dos seguintes valores:

#### **MQSVC\_CONTROL\_MANUAL**

O servidor de comandos não deve ser iniciado automaticamente

#### **MQSVC\_CONTROL\_Q\_MGR**

O servidor de comando deve ser iniciado automaticamente quando o gerenciador de filas for iniciado

Esse atributo não é suportado no z/OS

Para determinar o valor desse atributo, use o seletor MQIA\_CMD\_SERVER\_CONTROL com a chamada MQINQ.

### **ConfigurationEvent (MQLONG)**

Controla se os eventos de configuração são gerados

Para determinar o valor desse atributo, use o seletor MQIA\_CONFIGURATION\_EVENT com a chamada MQINQ..

O valor pode ser qualquer um dos seguintes valores:

#### **MQEVN\_DISABLED**

Relatório de eventos desativado.

#### **MQEVN\_ENABLED**

Relatório de eventos ativado

### **Multi CurrentQFileTamanho (MQLONG)**

O tamanho atual do arquivo de fila em megabytes, arredondado para o megabyte mais próximo.

<i>Tabela 558. Tipos de fila aos quais este atributo se aplica</i>				
Local	Modelo	Alias	Remoto	Cluster
X	X			

O valor para esse atributo de status da fila é qualquer que seja o tamanho da fila atualmente, arredondado para o megabyte mais próximo. Para uma nova fila com atributos padrão, o valor de **CurrentQFileSize** é 1.

O valor máximo desse atributo é 99.999.9999 MB e não há valor padrão para esse atributo.

### **Multi CurrentMaxQFileSize (MQLONG).**

O tamanho máximo atual para o qual o arquivo de fila pode crescer, arredondado para cima até o megabyte mais próximo, dado o tamanho do bloco atual em uso em uma fila

<i>Tabela 559. Tipos de fila aos quais este atributo se aplica</i>				
Local	Modelo	Alias	Remoto	Cluster
X	X			

O uso deste campo é duas vezes:

- Se você configurar **MaxQFileSize** como o valor padrão para o tamanho de bloco atual, **CurrentMaxQFileSize** mostrará o valor real que o valor padrão equivale.
- Se **CurrentMaxQFileSize** não corresponder **MaxQFileSize**, você saberá que a fila deve ser drenada para adotar uma granularidade maior.

**Nota:** Consulte [Modificando IBM MQ arquivos de fila](#) para obter mais informações sobre como alterar o tamanho dos arquivos de filas, o tamanho do bloco e a granularidade

O valor máximo desse atributo é 99,999,9999 MB e não há valor padrão. O valor é qualquer que seja o valor máximo configurado atualmente; para uma nova fila com os atributos padrão, o valor de **CurrentMaxQFileSize** é 2.088.960 MB.

### **DeadLetterQName (MQCHAR48)**

Este é o nome de uma fila definida no gerenciador de fila local como a fila de mensagens não entregues. As mensagens serão enviadas para esta fila se não puderem ser roteados para o seu destino correto

Por exemplo, as mensagens são colocadas nesta fila quando:



- Uma mensagem chega a um gerenciador de filas, destinado para uma fila que ainda não está definida nesse gerenciador de fila
- Uma mensagem chega em um gerenciador de filas, mas a fila para a qual ela está destinada não pode recebê-la porque, possivelmente:
  - A fila está cheia
  - As solicitações put são inibidas
  - O nó de envio não tem autoridade para colocar mensagens na fila

Aplicativos também podem colocar mensagens na fila de mensagens não entregues.

As mensagens de relatório são tratadas da mesma maneira que as mensagens ordinárias; se a mensagem de relatório não puder ser entregue à sua fila de destino (geralmente a fila especificada pelo campo *ReplyToQ* no descritor de mensagens da mensagem original), a mensagem de relatório será colocada na fila de mensagens não entregues (mensagens não entregues).

**Nota:** Mensagens que passaram seu tempo de expiração (consulte [MQMD-Campo de expiração](#)) **não** são transferidos para esta fila quando eles são descartados. No entanto, uma mensagem de relatório de expiração (MQRO\_EXPIRATION) ainda é gerada e enviada para a fila *ReplyToQ*, se solicitado pelo aplicativo de envio.

As mensagens não são colocadas na fila de mensagens não entregues (mensagens não entregues) quando o aplicativo que emitiu a solicitação put foi notificado de forma síncrona do problema por meio do código de razão retornado pela chamada MQPUT ou MQPUT1 (por exemplo, uma mensagem colocada em uma fila local para a qual as solicitações put são inibidas).

As mensagens na fila de mensagens não entregues (não entregues) às vezes têm seus dados da mensagem do aplicativo prefixados com uma estrutura MQDLH. Essa estrutura contém informações extras que indicam por que a mensagem foi colocada na fila de mensagens não entregues (mensagens não entregues). Consulte [“MQDLH - Cabeçalho de mensagens não entregues”](#) na página 361 para obter mais detalhes sobre essa estrutura.

Essa fila deve ser uma fila local, com um atributo **Usage** MQUS\_NORMAL.

Se um gerenciador de filas não suportar uma fila de mensagens não entregues (mensagens não entregues) ou uma não tiver sido definida, o nome estará em branco. Todos os gerenciadores de filas do IBM MQ suportam uma fila de mensagens não entregues, mas, por padrão, ela não está definida.

Se a fila de mensagens não entregues (mensagens não entregues) não for definida, completa ou inutilizável por alguma outra razão, uma mensagem que teria sido transferida para ela por um agente do canal de mensagens será retida na fila de transmissão.

Para determinar o valor desse atributo, use o seletor MQCA\_DEAD\_LETTER\_Q\_NAME com a chamada MQINQ. O comprimento desse atributo é fornecido por MQ\_Q\_NAME\_LENGTH.

### **DefClusterXmitQueueTipo (MQLONG)**

O atributo DefClusterXmitQueueTipo controla qual fila de transmissão é selecionada por padrão pelos canais do emissor de clusters para obter mensagens, para enviar as mensagens para os canais do receptor de clusters.

Os valores de **DefClusterXmitQueueType** são MQCLXQ\_SCTQ ou MQCLXQ\_CHANNEL.

#### **MQCLXQ\_SCTQ**

Todos os canais do emissor de clusters enviam mensagens a partir do SYSTEM.CLUSTER.TRANSMIT.QUEUE. O correlID de mensagens colocadas na fila de transmissão identifica para qual canal do emissor de clusters a mensagem é destinada.

SCTQ é configurado quando um gerenciador de filas é definido.

#### **MQCLXQ\_CHANNEL**

Cada canal do emissor de clusters envia mensagens a partir de uma fila de transmissão diferente. Cada fila de transmissão é criada como uma fila dinâmica permanente a partir da fila de modelos SYSTEM.CLUSTER.TRANSMIT.MODEL.QUEUE.

Se o atributo do gerenciador de filas, DefClusterXmitQueueType, for configurado como CHANNEL A configuração padrão será alterada para os canais do emissor de clusters que estão sendo associados às filas de transmissão de cluster individuais. As filas de transmissão são filas permanentes e dinâmicas criadas a partir da fila modelo SYSTEM . CLUSTER . TRANSMIT . MODEL . QUEUE. Cada fila de transmissão é associada a um canal do emissor de clusters. Conforme um canal do emissor de clusters atende uma fila de transmissão do cluster, a fila de transmissão contém mensagens apenas para um gerenciador de filas em um cluster. É possível configurar clusters para que cada gerenciador de filas em um cluster contenha apenas uma fila de clusters. Neste caso, o tráfego de mensagens de um gerenciador de filas para cada fila de clusters é transferido separadamente de mensagens para outras filas.

Para consultar o valor, chame MQINQ ou envie um comando PCF Inquire Queue Manager (MQCMD\_INQUIRE\_Q\_MGR), configurando o seletor MQIA\_DEF\_CLUSTER\_XMIT\_Q\_TYPE . Para alterar o valor, envie um comando PCF Change Queue Manager (MQCMD\_CHANGE\_Q\_MGR), configurando o seletor MQIA\_DEF\_CLUSTER\_XMIT\_Q\_TYPE .

### **Referências relacionadas**

[Alterar Gerenciador de Filas](#)

[Consultar Gerenciador de Filas](#)

“MQINQ-Consultar atributos do objeto.” na página 730

A chamada MQINQ retorna uma matriz de números inteiros e um conjunto de cadeias de caracteres que contêm os atributos de um objeto

### **DefXmitQName (MQCHAR48)**

Este é o nome da fila de transmissão que é usada para a transmissão de mensagens para gerenciadores de filas remotas, se não houver nenhuma outra indicação de qual fila de transmissão usar

Se não houver uma fila de transmissão padrão, o nome estará inteiramente em branco O valor inicial deste atributo está em branco

Para determinar o valor desse atributo, use o seletor MQCA\_DEF\_XMIT\_Q\_NAME com a chamada MQINQ.. O comprimento desse atributo é fornecido por MQ\_Q\_NAME\_LENGTH

### **DistLists (MQLONG)**

Indica se o gerenciador de filas locais suporta listas de distribuição nas chamadas MQPUT e MQPUT1 . Ele é um dos seguintes valores:

#### **MQDL\_SUPPORTED**

Listas de distribuição suportadas.

#### **MQDL\_NOT\_SUPPORTED**

Listas de distribuição não suportadas.

Para determinar o valor desse atributo, use o seletor MQIA\_DIST\_LISTS com a chamada MQINQ..

### **DNSGroup (MQCHAR18)**

Esse parâmetro não é mais utilizado. Consulte [O que mudou em IBM MQ 8.0.](#)

Esse atributo é suportado apenas no z/OS

Para determinar o valor desse atributo, use o seletor MQCA\_DNS\_GROUP com a chamadas MQINQ. O comprimento desse atributo é fornecido por MQ\_DNS\_GROUP\_NAME\_LENGTH

### **DNSWLM (MQLONG)**

Esse parâmetro não é mais utilizado. Consulte [O que mudou em IBM MQ 8.0.](#)

O valor é um dos seguintes:

#### **MQDNSWLM\_YES**

Esse valor pode ser visto em um gerenciador de filas migrado de uma versão anterior.. O valor é ignorado.

## **MQDNSWLM\_NO**

Este é o único valor suportado pelo gerenciador de filas.

Esse atributo é suportado apenas no z/OS

Para determinar o valor deste atributo, utilize o seletor MQIA\_DNS\_WLM com a chamada MQINQ.


## **ExpiryInterval (MQLONG)**

Isso indica a frequência com que o gerenciador de filas varre as filas procurando mensagens expiradas. É um intervalo de tempo em segundos no intervalo de 1 a 99 999 999 999 ou o valor especial a seguir:

### **MQEXPI\_OFF**

O gerenciador de filas não varre as filas procurando mensagens expiradas.

Para determinar o valor desse atributo, use o seletor MQIA\_EXPIRY\_INTERVAL com a chamada MQINQ

 Esse atributo é suportado apenas no z/OS

## **IGQPutAuthority (MQLONG)**

Esse atributo se aplica somente se o gerenciador de filas locais for membro de um grupo de filas compartilhadas. Indica o tipo de verificação de autoridade que é executada quando o agente de enfileiramento intragrupo local (agente IGQ) remove uma mensagem da fila de transmissão compartilhada e coloca a mensagem em uma fila local. O valor é um dos seguintes:

### **MQIGQPA\_DEFAULT**

O identificador de usuários verificado para autorização é o valor do campo *UserIdentifier* no MQMD *separado* associado à mensagem quando a mensagem está na fila de transmissão compartilhada. Este é o identificador de usuário do programa que colocou a mensagem na fila de transmissão compartilhada e geralmente é o mesmo identificador de usuário sob o qual o gerenciador de filas remotas está em execução.

Se o perfil RESLEVEL indicar que mais de um identificador de usuário será verificado, o identificador de usuário do agente IGQ local (*IGQUserId*) também será verificado.

### **MQIGQPA\_CONTEXT**

O identificador de usuários verificado para autorização é o valor do campo *UserIdentifier* no MQMD *separado* associado à mensagem quando a mensagem está na fila de transmissão compartilhada. Este é o identificador de usuário do programa que colocou a mensagem na fila de transmissão compartilhada e geralmente é o mesmo identificador de usuário sob o qual o gerenciador de filas remotas está em execução.

Se o perfil RESLEVEL indicar que mais de um identificador de usuário deve ser verificado, o identificador de usuário do agente IGQ local (*IGQUserId*) e o valor do campo *UserIdentifier* no MQMD *integrado* também serão verificados.. O último identificador de usuário é geralmente o identificador de usuário do aplicativo que originou a mensagem..

### **MQIGQPA\_ONLY\_IGQ**

O identificador de usuários verificado para autorização é o identificador de usuários do agente IGQ local (*IGQUserId*).

Se o perfil RESLEVEL indicar que mais de um identificador de usuário deve ser verificado, esse identificador de usuário será usado para todas as verificações..

### **MQIGQPA\_ALTERNATE\_OR\_IGQ**

O identificador de usuários verificado para autorização é o identificador de usuários do agente IGQ local (*IGQUserId*).

Se o perfil RESLEVEL indicar que mais de um identificador de usuário será verificado, o valor do campo *UserIdentifier* no MQMD *integrado* também será verificado. Esse identificador de usuário é geralmente o identificador de usuário do aplicativo que originou a mensagem..

Para determinar o valor desse atributo, use o seletor MQIA\_IGQ\_PUT\_AUTHORITY com a chamada MQINQ.



Esse atributo é suportado apenas no z/OS

### ***IGQUserId (MQLONG)***

Esse atributo é aplicável apenas se o gerenciador da fila local for um membro de um grupo de filas compartilhadas. Ele especifica o identificador de usuários associado ao agente de enfileiramento intragrupo local (agente IGQ). Esse identificador é um dos identificadores de usuário que podem ser verificados para autorização quando o agente IGQ coloca mensagens em filas locais. Os identificadores de usuário reais verificados dependem da configuração do atributo **IGQPutAuthority** e das opções de segurança externas.

Se *IGQUserId* estiver em branco, nenhum identificador de usuário será associado ao agente IGQ e a verificação de autorização correspondente não será executada (embora outros identificadores de usuário ainda possam ser verificados quanto à autorização).

Para determinar o valor desse atributo, use o seletor `MQCA_IGQ_USER_ID` com a chamada `MQINQ`. O comprimento desse atributo é fornecido por `MQ_USER_ID_LENGTH`.



Esse atributo é suportado apenas no z/OS

### ***InhibitEvent (MQLONG)***

Isso controla se eventos inibidos (Inhibit Get e Inhibit Put) são gerados. O valor é um dos seguintes:

#### **MQEVR\_DISABLED**

Relatório de eventos desativado.

#### **MQEVR\_ENABLED**

Relatório de eventos ativado.

Para obter mais informações sobre os eventos, consulte [Monitoramento de eventos](#).

Para determinar o valor desse atributo, use o seletor `MQIA_INIBIT_EVENT` com a chamada `MQINQ`.

No z/OS, não é possível usar a chamada `MQINQ` para determinar o valor desse atributo...

### ***IntraGroupqueuing (MQLONG)***

Esse atributo se aplica somente se o gerenciador de filas locais for membro de um grupo de filas compartilhadas. Ele indica se o enfileiramento intragrupo está ativado para o grupo de filas compartilhadas. O valor é um dos seguintes:

#### **MQIGQ\_DISABLED**

Todas as mensagens destinadas a outros gerenciadores de filas no grupo de filas compartilhadas são transmitidas usando os canais convencionais ...

#### **MQIGQ\_ENABLED**

As mensagens destinadas a outros gerenciadores de fila no grupo de filas compartilhadas são transmitidas usando a fila de transmissão compartilhada se a condição a seguir for satisfeita:

- O comprimento dos dados da mensagem mais o cabeçalho de transmissão não excede 63 KB (64 512 bytes)

É recomendado que um pouco mais de espaço do que o tamanho de `MQXQH` seja alocado para o cabeçalho de transmissão; a constante `MQ_MSG_HEADER_LENGTH` é fornecida para esse propósito.

Se essa condição não for satisfeita, a mensagem será transmitida usando canais convencionais.

**Nota:** Quando o enfileiramento intragrupo é ativado, a ordem de mensagens transmitidas usando a fila de transmissão compartilhada não é preservado em relação àquelas transmitidas usando canais convencionais.

Para determinar o valor desse atributo, use o seletor `MQIA_INTRA_GROUP_queuing` com a chamada `MQINQ`.



Esse atributo é suportado apenas no z/OS

### ***IPAddressVersion (MQLONG)***

Especifica qual versão de endereço IP, seja IPv4 ou IPv6, é usada

Esse atributo é relevante apenas para sistemas que executam IPv4 e IPv6 e afeta apenas os canais definidos como tendo um *TransportType* de MQXPY\_TCP quando uma das condições a seguir é verdadeira:

- O *ConnectionName* do canal é um nome do host que resolve para um endereço IPv4 e IPv6 e seu parâmetro **LocalAddress** não é especificado.
- Os *ConnectionName* e *LocalAddress* do canal são ambos nomes de host que são resolvidos para os endereços IPv4 e IPv6 .

O valor pode ser qualquer um dos seguintes valores:

#### **MQIPADDR\_IPV4**

IPv4 será usada.

#### **MQIPADDR\_IPV6**

IPv6 será usada.

Para determinar o valor desse atributo, use o seletor MQIA\_IP\_ADDRESS\_VERSION com a chamada MQINQ

### ***ListenerTimer (MQLONG)***

Este é o intervalo de tempo (em segundos) entre IBM MQ tentativas de reiniciar o listener se houver uma falha de APPC ou TCP/IP. O valor deve estar entre 5 e 9999, com um valor padrão de 60.

Esse atributo é suportado apenas no z/OS

Para determinar o valor desse atributo, use o seletor MQIA\_LISTENER\_TIMER com a chamada MQINQ..

### ***LocalEvent (MQLONG)***

Isso controla se os eventos de erro locais são gerados O valor é um dos seguintes:

#### **MQEVR\_DISABLED**

Relatório de eventos desativado.

#### **MQEVR\_ENABLED**

Relatório de eventos ativado

Para obter mais informações sobre os eventos, consulte [Monitoramento de eventos](#)

Para determinar o valor desse atributo, use o seletor MQIA\_LOCAL\_EVENT com a chamada MQINQ..

No z/OS, não é possível usar a chamada MQINQ para determinar o valor desse atributo...

### ***LoggerEvent (MQLONG)***

Isso controla se os eventos do registro de recuperação são gerados O valor é um dos seguintes:

#### **MQEVR\_DISABLED**

Relatório de eventos desativado.

#### **MQEVR\_ENABLED**

Relatório de eventos ativado

Para obter mais informações sobre os eventos, consulte [Monitoramento de eventos](#)

Para determinar o valor desse atributo, use o seletor MQIA\_LOGGER\_EVENT com a chamada MQINQ.



Esse atributo é suportado apenas no [Multiplataformas](#)

### ***LUGroupName (MQCHAR8)***

Este é o nome da LU genérica do listener da LU 6.2 que manipula as transmissões de entrada para o grupo de filas compartilhadas. Se você deixar esse nome em branco, não será possível usar esse listener.

Esse atributo é suportado apenas no z/OS.

Para determinar o valor desse atributo, use o seletor MQCA\_LU\_GROUP\_NAME com a chamada MQINQ. O comprimento desse atributo é fornecido por MQ\_LU\_NAME\_LENGTH.

### **LUName (MQCHAR8).**

Este é o nome da LU a ser usada para transmissões de LU de saída 6.2. Configure-o para a mesma LU que o listener usa para transmissões de entrada. Se você deixar esse nome em branco, a LU padrão APPC/MVS será usada; essa é uma variável, portanto, sempre configure LUName se estiver usando LU6.2.

Esse atributo é suportado apenas no z/OS.

Para determinar o valor desse atributo, use o seletor MQCA\_LU\_NAME com a chamada MQINQ. O comprimento desse atributo é fornecido por MQ\_LU\_NAME\_LENGTH.

### **LU62ARMSuffix (MQCHAR2)**

Este é o sufixo do SYS1.PARMLIB membro APPCPMxx, que nomeia LUADD para este inicializador de canais. O comando z/OS SET APPC=xx é emitido quando o ARM reinicia o inicializador de canais. Se você deixar esse nome em branco, nenhum SET APPC=xx será emitido.

Esse atributo é suportado apenas no z/OS.

Para determinar o valor desse atributo, use o seletor MQCA\_LU62\_ARM\_SUFFIX com a chamada MQINQ. O comprimento desse atributo é fornecido por MQ\_ARM\_SUFFIX\_LENGTH.

### **LU62Channels (MQLONG)**

Este é o número máximo de canais que podem ser atuais ou clientes que podem ser conectados, que usam o protocolo de transmissão da LU 6.2.

O valor deve estar no intervalo de 0 a 9999, com um valor padrão de 200. Se você configurar isso como zero, o protocolo de transmissão LU 6.2 não será usado.

Esse atributo é suportado apenas no z/OS.

Para determinar o valor deste atributo, use o seletor MQIA\_LU62\_CHANNELS com a chamada MQINQ.


### **Canais MaxActive(MQLONG)**

Este atributo é o número máximo de canais que podem estar *ativos* a qualquer momento.

O padrão é o valor especificado para o atributo MaxChannels.

Para z/OS, o valor deve estar no intervalo de 1 a 9 999.

Para todas as outras plataformas, o valor padrão é 999 999 999, o que significa que o número de canais ativos é ilimitado ou pode ser configurado para um número real para impor um limite.

 Você não deve alterar o valor de **MaxActiveChannels** em IBM MQ Appliance. Se desejar limitar o número máximo de canais do cliente, use os atributos MAXINST e MAXINSTC por canal nas definições de canal SVRCONN para definir limites para cada canal SVRCONN, consulte [Configuração do gerenciador de filas no IBM MQ Appliance](#) na documentação IBM MQ Appliance.

O parâmetro **MaxActiveChannels** é um atributo do Gerenciador de Filas somente em z/OS. Nas outras plataformas, **MaxActiveChannels** é um atributo no arquivo qm.ini. Consulte [Sub-rotinas do arquivo de configuração para o enfileiramento distribuído](#) para obter informações sobre como configurar o atributo **MaxActiveChannels** em outras plataformas.

Para determinar o valor desse atributo, use o seletor MQIA\_ACTIVE\_CHANNELS com a chamada MQINQ.

### **Conceitos relacionados**

[Estados de Canais](#)

## **MaxChannels (MQLONG)**

Este atributo é o número máximo de canais que podem ser *atuais* (incluindo canais de conexão do servidor com clientes conectados)

Para z/OS, o valor deve estar no intervalo de 1 a 9 999, com um valor padrão de 200.

**MQ Appliance** Para o IBM MQ Appliance, o valor padrão é 999 999 999 e não deve ser mudado. Se desejar limitar o número máximo de canais do cliente, use os atributos MAXINST e MAXINSTC por canal nas definições de canal SVRCONN para definir limites para cada canal SVRCONN, consulte [Configuração do gerenciador de filas no IBM MQ Appliance](#) na documentação IBM MQ Appliance .

Um sistema que está ocupado atendendo conexões da rede pode precisar de um número maior do que a configuração padrão. Determine o valor que está correto para seu ambiente, idealmente, observando o comportamento de seu sistema durante o teste.

Para todas as outras plataformas, o valor padrão é 100. É possível configurar **MaxChannels** para um valor diferente para limitar o número máximo de canais atuais se necessário.

O parâmetro **MaxChannels** é um atributo do Gerenciador de Filas somente em z/OS Nas outras plataformas, **MaxChannels** é um atributo no arquivo `qm.ini` . Consulte [Sub-rotinas do arquivo de configuração para o enfileiramento distribuído](#) para obter informações sobre como configurar o atributo **MaxChannels** em outras plataformas

Para determinar o valor desse atributo, use o seletor MQIA\_MAX\_CHANNELS com a chamada **MQINQ** .

### **Conceitos relacionados**

[Estados de Canais](#)

## **MaxHandles (MQLONG)**

Este é o número máximo de identificadores abertos que qualquer tarefa pode utilizar simultaneamente Cada chamada MQOPEN bem-sucedida para uma única fila (ou para um objeto que não seja uma fila) usa uma manipulação de um identificador. Essa manipulação torna-se disponível para reutilização quando o objeto é encerrado No entanto, quando uma lista de distribuição é aberta, cada fila na lista de distribuição é alocada um identificador separado e para que a chamada MQOPEN use tantos identificadores quanto houver filas na lista de distribuição. Isso deve ser considerado ao decidir sobre um valor adequado para *MaxHandles*.

A chamada MQPUT1 executa uma chamada MQOPEN como parte de seu processamento; como resultado, MQPUT1 usa tantas manipulações como MQOPEN, mas as manipulações são usadas apenas para a duração da própria chamada MQPUT1

Em z/OS, *task* significa uma tarefa CICS , uma tarefa MVS ou uma região dependente do IMS .

O valor está no intervalo de 1 a 999 999 999. O valor padrão é determinado pelo ambiente:

- No z/OS, o valor padrão é 100.
- Em todos os outros ambientes, o valor padrão é 256..

Para determinar o valor desse atributo, use o seletor MQIA\_MAX\_CONSIGO com a chamada MQINQ.

## **Comprimento de MaxMsg(MQLONG)**

Este é o comprimento da mensagem *física* mais longa que o gerenciador de filas pode manipular No entanto, como o atributo do gerenciador de fila **MaxMsgLength** pode ser configurado independentemente do atributo da fila **MaxMsgLength** , a mensagem física mais longa que pode ser colocada em uma fila é o menor desses dois valores

Se o gerenciador de filas suportar segmentação, um aplicativo poderá colocar uma mensagem lógica maior que o menor dos dois atributos **MaxMsgLength** , mas somente se o aplicativo especificar o sinalizador MQMF\_SEGMENTATION\_ALLOWED no MQMD. Se esse sinalizador for especificado, o limite superior para o comprimento de uma mensagem lógica é 999 999 999 bytes, mas geralmente restrições de recursos impostas pelo sistema operacional, ou pelo ambiente no qual o aplicativo está em execução, resultam em um limite inferior.



O limite inferior para o atributo **MaxMsgLength** é 32 KB (32 768 bytes). O limite superior é de 100 MB (104 857 600 bytes).

Para determinar o valor desse atributo, use o seletor MQIA\_MAX\_MSG\_LENGTH com a chamada MQINQ..

### **MaxPriority (MQLONG)**

Esta é a prioridade máxima da mensagem suportada pelo gerenciador de fila. As prioridades variam de zero (mais baixo) a *MaxPriority* (mais alto).

Para determinar o valor desse atributo, use o seletor MQIA\_MAX\_PRIORITY com a chamadas MQINQ.

### **Comprimento de MaxProperties(MQLONG)**

Isso é utilizado para controlar o tamanho das propriedades que podem fluir com uma mensagem. Isso inclui o nome da propriedade em bytes e o tamanho do valor da propriedade também em bytes.

Para determinar o valor desse atributo, use o seletor MQIA\_MAX\_PROPERTIES\_LENGTH com a chamada MQINQ

### **Multi MaxQFileTamanho (MQLONG)**

O tamanho máximo, em megabytes, que um arquivo de fila pode crescer.

Tabela 560. Tipos de fila aos quais este atributo se aplica				
Local	Modelo	Alias	Remoto	Cluster
X	X			

É possível que um arquivo de fila exceda o tamanho máximo, se estiver configurado para um valor inferior ao tamanho do arquivo de fila atual. Se isso acontecer, o arquivo de fila não aceita mais novas mensagens, mas permite que mensagens existentes sejam consumidas. Quando o tamanho do arquivo de fila estiver abaixo do valor configurado, novas mensagens poderão ser colocadas na fila.

**Nota:** Esta figura pode diferir do valor do atributo configurado na fila, porque internamente o gerenciador de filas pode precisar usar um tamanho de bloco maior para atingir o tamanho escolhido... Consulte [Modificando IBM MQ arquivos de fila](#) para obter mais informações sobre como alterar o tamanho dos arquivos de filas, o tamanho do bloco e a granularidade

Quando a granularidade precisar mudar porque esse atributo foi aumentado, a mensagem de aviso AMQ7493W Granularidade mudada será gravada nos logs AMQERR. Isso fornece uma indicação de que você precisa planejar que a fila seja esvaziada, para que o IBM MQ adote a nova granularidade.

O valor máximo desse atributo é 267.386.880 MB e o valor padrão e o valor migrado é 2.088.960 MB, que é o máximo atual para uma fila com uma granularidade igual a 512.

Para determinar o valor desse atributo, use o seletor MQIA\_MAX\_Q\_FILE\_SIZE com a chamada MQINQ

### **MaxUncommittedMsgs (MQLONG)**

Este é o número máximo de mensagens não consolidadas que podem existir em uma unidade de trabalho. O número de mensagens não confirmadas é a soma do seguinte, desde o início da unidade de trabalho atual:

- Mensagens colocadas pelo aplicativo com a opção MQPMO\_SYNCPOINT
- Mensagens recuperadas pelo aplicativo com a opção MQGMO\_SYNCPOINT
- Mensagens do acionador e mensagens de relatório COA geradas pelo gerenciador de filas para mensagens enviadas com a opção MQPMO\_SYNCPOINT
- Mensagens de relatório COD geradas pelo gerenciador de filas para mensagens recuperadas com a opção MQGMO\_SYNCPOINT

As seguintes mensagens não são contadas como não confirmadas:

- Mensagens colocadas ou recuperadas pelo aplicativo fora de uma unidade de trabalho



- Mensagens do acionador ou mensagens de relatório COA/COD geradas pelo gerenciador de filas como resultado de mensagens colocadas ou recuperadas fora de uma unidade de trabalho.
- Mensagens de relatório de expiração geradas pelo gerenciador de filas (mesmo se a chamada estiver causando a mensagem de relatório de expiração especificada MQGMO\_SYNCPOINT)
- Mensagens de evento geradas pelo gerenciador de filas (mesmo se a chamada estiver causando a mensagem do evento especificada MQPMO\_SYNCPOINT ou MQGMO\_SYNCPOINT).

**Nota:**

1. As mensagens de relatório de exceção são geradas pelo Message Channel Agent (MCA) ou pelo aplicativo e são tratadas da mesma maneira que as mensagens ordinárias colocadas ou recuperadas pelo aplicativo.
2. Quando uma mensagem ou segmento é colocado com a opção MQPMO\_SYNCPOINT, o número de mensagens não confirmadas é incrementado em um, independentemente de quantas mensagens físicas realmente resultam da colocação. (Mais de uma mensagem física pode resultar se o gerenciador de filas deve subdividir a mensagem ou segmento.)
3. Quando uma lista de distribuição for colocada com a opção MQPMO\_SYNCPOINT, o número de mensagens não confirmados será incrementada em um *para cada mensagem física gerada*. Isso pode ser tão pequeno quanto um ou tão grande quanto o número de destinos na lista de distribuição.

O limite inferior desse atributo é 1; o limite superior é 999 999 999. O valor-padrão é 10000.

Para determinar o valor desse atributo, use o seletor MQIA\_MAX\_UNCOMMITTED\_MSGS com a chamada MQINQ..

### ***MQIAccounting (MQLONG)***

Isso controla a coleta de informações de contabilidade para dados MQI.

O valor é um dos seguintes:

**MQMON\_ON**

Coletar dados de contabilidade da API.

**MQMON\_OFF**

Não colete dados de contabilidade da API. Esse é o valor-padrão.

Se você configurar o atributo do gerenciador de filas ACCTCONO como ENABLED, esse valor poderá ser substituído para conexões individuais usando o campo Opções na estrutura MQCNO. As mudanças nesse valor são efetivas apenas para conexões com o gerenciador de filas que ocorrem após a mudança no atributo.

Esse atributo é suportado apenas no Multiplataformas

Para determinar o valor desse atributo, use o seletor MQIA\_ACCOUNTING\_MQI com a chamada MQINQ.

### ***MQIStatistics (MQLONG)***

Isso controla a coleção de informações de monitoramento de estatísticas para o gerenciador de fila

O valor é um dos seguintes:

**MQMON\_ON**

Coletar estatísticas de MQI.

**MQMON\_OFF**

Não coletar estatísticas de MQI. Esse é o valor-padrão.

Esse atributo é suportado apenas no Multiplataformas

Para determinar o valor desse atributo, use o seletor MQIA\_STATISTICS\_MQI com a chamada MQINQ.

### ***MsgMarkBrowseInterval (MQLONG).***

Intervalo de tempo em milissegundos após o qual o gerenciador de filas pode remover automaticamente a marca das mensagens de procura.

Este é um intervalo de tempo (em milissegundos) após o qual o gerenciador de filas pode remover automaticamente a marca de mensagens de procura...

Este atributo descreve o intervalo de tempo para o qual as mensagens que foram marcadas como navegadas por uma chamada para MQGET, usando a opção de mensagem get MQGMO\_MARK\_BROWSE\_CO\_OP, devem permanecer marcadas como navegadas.

O gerenciador de filas pode desmarcar automaticamente as mensagens procuradas que foram marcadas como procuradas para o conjunto de identificadores de cooperação quando tiverem sido marcadas para mais desse intervalo aproximado.

Isso não afeta o estado de nenhuma mensagem marcada como procura, que foi obtida por uma chamada para MQGET, usando a opção obter mensagem MQGMO\_MARK\_BROWSE\_HANDLE.

O valor máximo é 999 999 999 e o valor padrão é 5000. Um valor especial de -1 para *MsgMarkBrowseInterval* representa um intervalo de tempo ilimitado.



**Atenção:** Esse valor não deve estar abaixo do padrão de 5000

Para determinar o valor desse atributo, use o seletor MQIA\_MSG\_MARK\_BROWSE\_INTERVAL com a chamada MQINQ..

### ***OutboundPortMáx. (MQLONG)***

Esse é o número de porta mais alto no intervalo, definido por OutboundPortMin e OutboundPortMax, de números de porta a serem usados para ligar os canais de saída

O valor é um número inteiro no intervalo de 0 a 65535 e deve ser igual ou maior que o valor mínimo de OutboundPort. O valor padrão é 0.

Esse atributo é suportado apenas no z/OS

Para determinar o valor desse atributo, use o seletor MQIA\_OUTBOUND\_PORT\_MAX com a chamada MQINQ

### ***OutboundPortMin (MQLONG)***

Esse é o número de porta mais baixo no intervalo, definido por OutboundPortMin e OutboundPortMax, de números de porta a serem usados para ligar canais de saída

O valor é um número inteiro no intervalo de 0 a 65535 e deve ser igual ou menor que o valor Máximo de OutboundPort. O valor padrão é 0.

Esse atributo é suportado apenas no z/OS

Para determinar o valor desse atributo, use o seletor MQIA\_OUTBOUND\_PORT\_MIN com a chamada MQINQ

### ***PerformanceEvent (MQLONG)***

Isso controla se os eventos relacionados ao desempenho são gerados Ele é um dos seguintes valores:

#### **MQEVR\_DISABLED**

Relatório de eventos desativado.

#### **MQEVR\_ENABLED**

Relatório de eventos ativado

Para obter mais informações sobre os eventos, consulte [Monitoramento de eventos](#)

Para determinar o valor desse atributo, use o seletor MQIA\_PERFORMANCE\_EVENT com a chamada MQINQ..

### ***Plataforma (MQLONG)***

Isso indica o sistema operacional no qual o gerenciador de filas está em execução:

#### **MQPL\_AIX**

AIX (mesmo valor que MQPL\_UNIX)

**MQPL\_APPLIANCE**

IBM MQ Appliance

**MQPL\_MVS**

z/OS (mesmo valor de MQPL\_ZOS).

**MQPL\_OS390**

z/OS (mesmo valor de MQPL\_ZOS).

**MQPL\_OS400**

IBM i.

**MQPL\_UNIX**

UNIX.

**MQPL\_WINDOWS\_NT**

Windows .

**MQPL\_ZOS**

z/OS.

Para determinar o valor desse atributo, use o seletor MQIA\_PLATFORM com a chamada MQINQ

***PubSubNPInputMsg***

Se descartar ou manter uma mensagem de entrada não entregue.

O valor é um dos seguintes:

**MQUNDELIVERED\_DISCARD**

Mensagens de entrada não persistentes poderão ser descartadas se não puderem ser processadas.

Esse é o valor-padrão.

**MQUNDELIVERED\_KEEP**

Mensagens de entrada não persistentes não serão descartadas se elas não puderem ser processadas.

Nessa situação, a interface de publicação / assinatura enfileirada continuará a tentar novamente o processo em intervalos apropriados e não continuará processando mensagens subsequentes

Para determinar o valor desse atributo, use o seletor MQIA\_PUBSUB\_NP\_MSG com a chamada MQINQ..

***PubSubNPResponse (MQLONG)***

Controla o comportamento das mensagens de resposta não entregues

O valor é um dos seguintes:

**MQUNDELIVERED\_NORMAL**

Respostas não persistentes que não podem ser colocadas na fila de respostas são colocadas na fila de mensagens não entregues, se não puderem ser colocadas na fila de mensagens não entregues, elas serão descartadas.

**MQUNDELIVERED\_SAFE**

Respostas não persistentes que não podem ser colocadas na fila de resposta são colocadas na fila de devoluções. Se a resposta não puder ser configurada e não puder ser colocada no DLQ, a interface de publicação / assinatura enfileirada retrocederá a operação atual e, em seguida, tentará novamente em intervalos apropriados e não continuará processando mensagens subsequentes

**MQUNDELIVERED\_DISCARD**

Respostas não persistentes não são colocadas na fila de resposta são descartadas.

Este é o valor padrão para novos gerenciadores de filas.

**MQUNDELIVERED\_KEEP**

As respostas não persistentes não são colocadas na fila de devoluções ou descartados Em vez disso, a interface de publicação / assinatura enfileirada retornará a operação atual e, em seguida, tentará novamente em intervalos apropriados.

Para determinar o valor desse atributo, use o seletor MQIA\_PUBSUB\_NP\_RESP com a chamada MQINQ

## Valor padrão para gerenciadores de filas migrados.

Se o gerenciador de filas tiver sido migrado do IBM MQ V6.0, o valor inicial desse atributo dependerá dos valores de *DiscardNonPersistentResponse* e *DLQNonPersistentResponse* antes da migração, conforme mostrado na seguinte tabela.

		Resposta DLQNonPersistent		
		Sim	No	Não definido
DiscardNonPersistentResponse	Sim	MQUNDELIVERED_NORMAL	MQUNDELIVERED_DISCARD	MQUNDELIVERED_NORMAL
	Não	MQUNDELIVERED_SAFE	MQUNDELIVERED_KEEP	MQUNDELIVERED_SAFE
	Not set	Se SyncPointPersistent = No, MQUNDELIVERED_SAFE else MQUNDELIVERED_NORMAL	Se SyncPointPersistent = No, MQUNDELIVERED_KEEP else MQUNDELIVERED_DISCARD	Se SyncPointPersistent = No, MQUNDELIVERED_SAFE else MQUNDELIVERED_NORMAL

### **PubSubMaxMsgRetryCount (MQLONG)**

O número de novas tentativas ao processar uma mensagem de comando com falha no ponto de sincronização.

O valor é um dos seguintes:

**0 - 999 999 999**

O valor padrão é 5 segundos.

Para determinar o valor desse atributo, use o seletor MQIA\_PUBSUB\_MAXMSG\_RETRY\_COUNT com a chamada MQINQ.

### **PubSubSyncPoint (MQLONG)**

Se apenas mensagens persistentes ou todas as mensagens são processadas sob o ponto de sincronização

O valor é um dos seguintes:

#### **MQSYNCPOINT\_IFPER**

Isso faz a interface de publicação / assinatura enfileirada receber mensagens não persistentes fora do ponto de sincronização. Se o daemon receber um ponto de sincronização externo à publicação, o daemon encaminhará a publicação para assinantes conhecidos para ele fora do ponto de sincronização.

Esse é o valor-padrão.

#### **MQSYNCPOINT\_YES**

Isso faz a interface de publicação / assinatura enfileirada receber todas as mensagens sob o ponto de sincronização.

Para determinar o valor deste atributo, utilize o seletor MQIA\_PUBSUB\_SYNC\_PT com a chamada MQINQ.

### **Modo PubSub(MQLONG)**

Se o mecanismo de publicação / assinatura e a interface de publicação / assinatura enfileirada estão em execução, permitindo, portanto, que os aplicativos publiquem / assinem usando a interface de programação de aplicativos e as filas que estão sendo monitorados pela interface de publicação / assinatura enfileirada

O valor é um dos seguintes:

#### **MQPSM\_COMPAT**

O mecanismo de publicação/assinatura está em execução. Portanto, é possível publicar / assinar usando a interface de programação de aplicativos. A interface de publicação / assinatura enfileirada não está em execução, portanto, nenhuma mensagem que é colocada nas filas que são monitoradas pela interface de publicação / assinatura enfileirada é executada. Essa configuração é usada para compatibilidade com WebSphere Message Broker V6 ou versões anteriores usando esse gerenciador de filas, porque ele deve ler as mesmas filas a partir das quais a interface de publicação / assinatura enfileirada normalmente lê.

## **MQPSM\_DISABLED**

O mecanismo de publicação/assinatura e a interface de publicação/assinatura enfileirada não estão em execução. Portanto, não é possível publicar / assinar usando a interface de programação de aplicativos.. Quaisquer mensagens de publicação / assinatura que são colocadas nas filas que são monitoradas pela interface de publicação / assinatura enfileirada não têm ação.

## **MQPSM\_ENABLED**

O mecanismo de publicação/assinatura e a interface de publicação/assinatura enfileirada estão em execução. Portanto, é possível publicar / assinar usando a interface de programação de aplicativos e as filas que estão sendo monitoradas pela interface de publicação / assinatura enfileiradas Este é o valor padrão inicial do gerenciador de filas.

Para determinar o valor desse atributo, use o seletor MQIA\_PUBSUB\_MODE com a chamada MQINQ.

## **QMGrDesc (MQCHAR64)**

Use este campo para um comentário descrevendo o gerenciador de filas O conteúdo do campo não é significativo para o gerenciador de filas, mas o gerenciador de filas pode requerer que o campo contenha apenas caracteres que possam ser exibidos... Ele não pode conter nenhum caractere nulo; se necessário, será preenchido à direita com espaços em branco. Em uma instalação DBCS, este campo pode conter caracteres DBCS (sujeito a um comprimento máximo de campo de 64 bytes).

**Nota:** Se esse campo contiver caracteres que não estão no conjunto de caracteres do gerenciador de filas (conforme definido pelo atributo do gerenciador de filas **CodedCharSetId**), esses caracteres poderão ser convertidos incorretamente se esse campo for enviado para outro gerenciador de filas..

- No z/OS, o valor padrão é o nome do produto e o número da versão
- Em todos os outros ambientes, o valor padrão é espaços em branco.

Para determinar o valor desse atributo, use o seletor MQCA\_Q\_MGR\_DESC com a chamada MQINQ.. O comprimento desse atributo é fornecido por MQ\_Q\_MGR\_DESC\_LENGTH

## **QMGrIdentifier (MQCHAR48)**

Este é um nome exclusivo gerado internamente para o gerenciador de filas

Para determinar o valor desse atributo, use o seletor MQCA\_Q\_MGR\_IDENTIFIER com a chamada MQINQ O comprimento desse atributo é fornecido por MQ\_Q\_MGR\_IDENTIFIER\_LENGTH

Esse atributo é suportado nos seguintes ambientes:

-  AIX
-  IBM i
-  Linux
-  Windows
-  z/OS

e clientes IBM MQ conectados a esses sistemas.

## **QMGrName (MQCHAR48)**

Este é o nome do gerenciador de fila local, ou seja o nome do gerenciador de fila ao qual o aplicativo está conectado

Os primeiros 12 caracteres do nome são usados para construir um identificador de mensagem exclusivo (consulte [MQMD-campo MsgId](#) ). Portanto, os gerenciadores de filas que podem intercomunicar devem ter nomes que diferem nos primeiros 12 caracteres, para que os identificadores de mensagens sejam exclusivos na rede do gerenciador de filas.


No z/OS, o nome é o mesmo que o nome do subsistema, que é limitado a 4 caracteres não em branco.

Para determinar o valor desse atributo, use o seletor MQCA\_Q\_MGR\_NAME com a chamada MQINQ. O comprimento desse atributo é fornecido por MQ\_Q\_MGR\_NAME\_LENGTH

### **QSGName (MQCHAR4)**

Este é o nome do grupo de filas compartilhadas ao qual o gerenciador de fila local pertence.. Se o gerenciador de filas locais não pertencer a um grupo de filas compartilhadas, o nome estará em branco

Para determinar o valor desse atributo, use o seletor MQCA\_QSG\_NAME com a chamada MQINQ. O comprimento deste atributo é fornecido por MQ\_QSG\_NAME\_LENGTH

 Esse atributo é suportado apenas no z/OS

### **QueueAccounting (MQLONG)**

Isso controla a coleta de informações de contabilidade para filas.

O valor é um dos seguintes:

#### **MQMON\_NONE**

Não colete dados de contabilidade para filas, independentemente da configuração do atributo de contabilidade da fila ACCTQ. Esse é o valor-padrão.

#### **MQMON\_OFF**

Não colete dados de contabilidade para filas que especificam QMGR no atributo da fila ACCTQ.

#### **MQMON\_ON**

Colete dados de contabilidade para filas que especificam QMGR no atributo da fila ACCTQ.

As mudanças nesse valor são efetivas apenas para conexões com o gerenciador de filas que ocorrem após a mudança no atributo.

Para determinar o valor desse atributo, use o seletor MQIA\_ACCOUNTING\_Q com a chamada MQINQ..

### **QueueMonitoring (MQLONG)**

Especifica a configuração padrão para monitoramento on-line de filas.

Se o atributo da fila **QueueMonitoring** for configurado como MQMON\_Q\_MGR, esse atributo especifica o valor que é assumido pelo canal. O valor do pode ser:

#### **MQMON\_OFF**

A coleta de dados de monitoramento on-line está desativada Este é o valor padrão inicial do gerenciador de filas.

#### **MQMON\_NONE**

A coleção de dados de monitoramento on-line é desativada para filas, independentemente da configuração de seu atributo **QueueMonitoring**

#### **MQMON\_LOW**

A coleta de dados de monitoramento on-line é ativada, com uma baixa proporção de coleta.

#### **MQMON\_MEDIUM**

A coleta de dados de monitoramento on-line é ativada, com uma proporção moderada de coleta de dados

#### **MQMON\_HIGH**

A coleta de dados de monitoramento on-line é ativada, com uma alta proporção de coleta de dados

Para determinar o valor desse atributo, use o seletor MQIA\_MONITORING\_Q com a chamada MQINQ..

### **QueueStatistics (MQLONG).**

Isso controla a coleta de dados de estatísticas para filas

Ele é um dos seguintes valores:

## **MQMON\_NONE**

Não colete estatísticas de fila para filas, independentemente da configuração do atributo da fila **QueueStatistics** . Esse é o valor-padrão.

## **MQMON\_OFF**

Não colete dados estatísticos para filas que especificam Gerenciador de Filas no atributo da fila **QueueStatistics** .

## **MQMON\_ON**

Colete dados estatísticos para filas que especificam o Gerenciador de Filas no atributo de fila **QueueStatistics**

Para determinar o valor deste atributo, use o seletor MQIA\_STATISTICS\_Q com a chamada MQINQ.

## **ReceiveTimeout (MQLONG)**

Isso especifica quanto tempo um canal TCP/IP aguarda para receber dados, incluindo pulsações, de seu parceiro antes de retornar ao estado inativo. Ele se aplica apenas aos canais de mensagens e não aos canais MQI

O significado exato de ReceiveTimeout é alterado pelo valor especificado no Tipo ReceiveTimeout. ReceiveTimeoutO tipo pode ser configurado para um dos seguintes:

- MQRCVTIME\_EQUAL-este valor é o número em segundos que o canal deve aguardar. Especifique um valor no intervalo de 0 a 999999
- MQRCVTIME\_ADD-esse valor é o número em segundos a ser incluído no HBINT negociado e determina quanto tempo um canal espera. Especifique um valor no intervalo de 1 a 999999
- MQRCVTIME\_MULTIPLY-esse valor é um multiplicador a ser aplicado ao HBINT negociado Especifique um valor 0 ou um valor no intervalo de 2 a 99.

O valor padrão é 0.

Configure o tipo ReceiveTimeoutpara MQRCVTIME\_MULTIPLY ou MQRCVTIME\_EQUAL e ReceiveTimeout para 0, para parar um canal de atingir o tempo limite de sua espera para receber dados de seu parceiro

Esse atributo é suportado apenas no z/OS

Para determinar o valor desse atributo, use o seletor MQIA\_RECEIVE\_TIMEOUT com a chamada MQINQ..

## **ReceiveTimeoutMín. (MQLONG)**

Este é o tempo mínimo, em segundos, que um canal TCP/IP espera para receber dados, incluindo pulsações, de seu parceiro, antes de retornar ao estado inativo.

Ele se aplica apenas aos canais de mensagem, não aos canais MQI O valor deve estar no intervalo de 0 a 999999, com um padrão 0.

Se você usar o Tipo ReceiveTimeoutpara especificar que o tempo de espera do canal TCP/IP deve ser calculado com relação ao valor negociado de HBINT e o valor resultante for menor que o valor desse parâmetro, esse valor será usado no lugar.

Esse atributo é suportado apenas no z/OS

Para determinar o valor desse atributo, use o seletor MQIA\_RECEIVE\_TIMEOUT\_MIN com a chamada MQINQ

## **ReceiveTimeoutTipo (MQLONG)**

Este é o qualificador, aplicado a ReceiveTimeout para definir quanto tempo um canal TCP/IP aguarda para receber dados, incluindo pulsações, de seu parceiro, antes de retornar ao estado inativo. Ele se aplica apenas aos canais de mensagem, não aos canais MQI

O valor é um dos seguintes:

**MQRCTIME\_MULTIPLY**

ReceiveTimeout é um multiplicador para aplicar ao valor HBINT negociado para determinar quanto tempo um canal aguarda. Esse é o valor-padrão.

**MQRCTIME\_ADD**

ReceiveTimeout é um valor, em segundos, para incluir no valor HBINT negociado para determinar quanto tempo um canal espera.

**MQRCTIME\_EQUAL**

ReceiveTimeout é um valor, em segundos, que o canal aguarda.

Para parar um canal atingindo o tempo limite de sua espera para receber dados de seu parceiro, configure ReceiveTimeoutTipo para MQRCTIME\_MULTIPLY ou MQRCTIME\_EQUAL e ReceiveTimeout para 0.

Esse atributo é suportado apenas no z/OS

Para determinar o valor desse atributo, use o seletor MQIA\_RECEIVE\_TIMEOUT\_TYPE com a chamada MQINQ

**RemoteEvent (MQLONG)**

Isso controla se os eventos de erro remotos são gerados Ele é um dos seguintes valores:

**MQEVN\_DISABLED**

Relatório de eventos desativado.

**MQEVN\_ENABLED**

Relatório de eventos ativado

Para obter mais informações sobre os eventos, consulte [Monitoramento de eventos](#)

Para determinar o valor desse atributo, use o seletor MQIA\_REMOTE\_EVENT com a chamada MQINQ

**RepositoryName (MQCHAR48)**

Este é o nome de um cluster para o qual este gerenciador de filas fornece um serviço do gerenciador do repositório Se o gerenciador de filas fornecer esse serviço para mais de um cluster, *RepositoryNameList* especificará o nome de um objeto de lista de nomes que identifica os clusters e *RepositoryName* estará em branco. Pelo menos um de *RepositoryName* e *RepositoryNameList* deve estar em branco

Para determinar o valor desse atributo, use o seletor MQCA\_REPOSITORY\_NAME com a chamada MQINQ.. O comprimento desse atributo é fornecido por MQ\_Q\_MGR\_NAME\_LENGTH

**RepositoryNameList (MQCHAR48)**

Este é o nome de um objeto de lista de nomes que contém os nomes de clusters para os qual este gerenciador de fila fornece um serviço de gerenciador de repositório Se o gerenciador de filas fornecer esse serviço para apenas um cluster, o objeto da lista de nomes conterá apenas um nome Como alternativa, *RepositoryName* pode ser usado para especificar o nome do cluster, em cujo caso, *RepositoryNameList* está em branco Pelo menos um de *RepositoryName* e *RepositoryNameList* deve estar em branco

Para determinar o valor desse atributo, use o seletor MQCA\_REPOSITORY\_NAMELIST com a chamada MQINQ.. O comprimento desse atributo é fornecido por MQ\_NAMELIST\_NAME\_LENGTH

**ScyCase(MQCHAR8)**

Especifica se o gerenciador de filas suporta nomes de perfis de segurança em letras maiúsculas ou em letras maiúsculas.

O valor é um dos seguintes:

**MQSCYC\_UPPER**


Os nomes do perfil de segurança devem estar em letras maiúsculas.



## **MQSCYC\_MIXED**

Os nomes do perfil de segurança podem estar em maiúsculas ou em letras maiúsculas e minúsculas.

As mudanças nesse atributo entram em vigor quando um comando Atualizar Segurança é executado com *SecurityType* (*MQSECTYPE\_CLASSES*) especificado.

 Esse atributo é suportado apenas no z/OS

Para determinar o valor desse atributo, use o seletor MQIA\_SECURITY\_CASE com a chamada MQINQ..

## **SharedQMgrNome (MQLONG)**

Isso especifica se o *ObjectQmgrName* deve ser usado ou tratado como o gerenciador de filas locais em uma chamada MQOPEN, para uma fila compartilhada, quando o *ObjectQmgrName* é o de outro gerenciador de filas no grupo de filas compartilhadas

O valor pode ser qualquer um dos seguintes valores:

### **MQSQQM\_USE**

O *ObjectQmgrName* é usado e a fila de transmissão apropriada é aberta

### **MQSQQM\_IGNORE**

Se a fila de destino for compartilhada e o *ObjectQmgrName* for o de um gerenciador de filas no mesmo grupo de filas compartilhadas, a abertura será executada localmente.

Esse atributo é válido apenas em z/OS

Para determinar o valor desse atributo, use o seletor MQIA\_SHARED\_Q\_Q\_MGR\_NAME com a chamada MQINQ.

## **SPLCAP**

Indica se recursos de segurança do Advanced Message Security estão disponíveis para um gerenciador de fila.

### **MQCAP\_SUPPORTED**

Este é o valor padrão se o componente AMS estiver instalado para a instalação sob a qual o gerenciador de filas está em execução.

### **MQCAP\_NOT\_SUPPORTED**

## **SSLEvent (MQLONG)**

Isso especifica se eventos TLS são gerados.

Ele é um dos seguintes valores:

### **MQEVR\_ENABLED**

Gerar eventos TLS, conforme a seguir:

MQRC\_CHANNEL\_SSL\_ERROR

### **MQEVR\_DISABLED**

Não gerar eventos TLS; este é o valor padrão.

Para determinar o valor desse atributo, use o seletor MQIA\_SSL\_EVENT com a chamada MQINQ.

## **SSLFIPSRequired (MQLONG)**

**Nota:** No AIX, Linux, and Windows, IBM MQ fornece conformidade FIPS 140-2 por meio do módulo criptográfico IBM Crypto for C (ICC) . O certificado deste módulo foi movido para o status Histórico. Os clientes devem visualizar o [IBM Crypto for C \(ICC\) certificado](#) e estar ciente de qualquer aviso fornecido pelo NIST Um módulo FIPS 140-3 de substituição está atualmente em andamento e seu status pode ser visualizado procurando por ele na [NIST CMVP modules in process list](#).

Isso permite especificar que apenas algoritmos certificados por FIPS devem ser usados se a criptografia for executada no IBM MQ, em vez de no hardware de criptografia. Se o hardware de criptografia for configurado, os módulos de criptografia usados serão aqueles módulos fornecidos pelo produto

de hardware; esses módulos podem ou não ser certificados por FIPS para um determinado nível, dependendo do produto de hardware em uso

O valor é um dos seguintes valores:

#### **MQSSL\_FIPS\_NO**

Use qualquer CipherSpec suportado na plataforma em uso. Esse valor é o valor padrão.

#### **MQSSL\_FIPS\_YES**

Use somente algoritmos criptográficos certificados por FIPS nos CipherSpecs permitidos em todas as conexões TLS de e para esse gerenciador de fila

Esse parâmetro é válido apenas nas plataformas z/OS, AIX, Linux, and Windows

Para determinar o valor desse atributo, use o seletor MQIA\_SSL\_FIPS\_REQUIRED com a chamada MQINQ..

#### **Tarefas relacionadas**

Especificando que Apenas CipherSpecs Certificados por FIPS São Usados no Tempo de Execução no Cliente de MQI

#### **Referências relacionadas**

Federal Information Processing Standards (FIPS) para AIX, Linux, and Windows

#### ***SSLKeyReset(MQLONG).***

Isso especifica quando os agentes do canal de mensagens do canal TLS (MCAs) que iniciam a comunicação reconfiguram a chave secreta usada para criptografia no canal.

O valor representa o número total de bytes não criptografados que são enviados e recebidos no canal antes da chave secreta ser renegociada. O número de bytes inclui informações de controle enviadas pelo MCA.

O valor é um número no intervalo de 0 a 999 999 999, com um valor padrão de 0. Se você especificar uma contagem de reconfiguração de chave secreta TLS no intervalo de 1 byte a 32 KB, os canais TLS usarão uma contagem de reconfiguração de chave secreta de 32 KB. Isso é para evitar o custo de processamento de reconfigurações de chave excessivas que ocorreriam para pequenos valores de reconfiguração de chave secreta TLS..

A chave secreta é renegociada quando o número total de bytes não criptografados enviados e recebidos pelo MCA do canal inicial excede o valor especificado. Se as pulsações do canal estiverem ativadas, a chave secreta será renegociada antes que os dados sejam enviados ou recebidos após uma pulsação do canal ou quando o número total de bytes não criptografados exceder o valor especificado, o que ocorrer primeiro.

A contagem de bytes enviados e recebidos para renegociação inclui informações de controle enviadas e recebidas pelo MCA do canal e é reconfigurada sempre que ocorre uma renegociação.

Use um valor 0 para indicar que as chaves secretas nunca são renegociadas.

Para determinar o valor desse atributo, use o seletor MQIA\_SSL\_RESET\_COUNT com a chamada MQINQ.

#### ***Evento StartStop(MQLONG)***

Isso controla se eventos de início e parada são gerados. O valor é um dos seguintes:

#### **MQEVR\_DISABLED**

Relatório de eventos desativado.

#### **MQEVR\_ENABLED**

Relatório de eventos ativado

Para obter mais informações sobre os eventos, consulte Monitoramento de eventos

Para determinar o valor desse atributo, use o seletor MQIA\_START\_STOP\_EVENT com a chamada MQINQ..

#### ***StatisticsInterval (MQLONG)***

Especifica com que frequência (em segundos) gravar dados de monitoramento de estatísticas na fila de monitoramento.

O valor é um número inteiro no intervalo de 0 a 604800, com um valor padrão de 1800 (30 minutos).

Para determinar o valor desse atributo, use o seletor MQIA\_STATISTICS\_INTERVAL com a chamada MQINQ.

### ***SyncPoint (MQLONG)***

Isso indica se o gerenciador de filas locais suporta unidades de trabalho e sincronização com as chamadas MQGET, MQPUT e MQPUT1 .

#### **MQSP\_AVAILABLE**

Unidades de trabalho e ponto de sincronização disponíveis

#### **MQSP\_NOT\_AVAILABLE**

Unidades de trabalho e sincronização não disponíveis.

- Em z/OS , esse valor nunca é retornado.

Para determinar o valor desse atributo, use o seletor MQIA\_SYNCPOINT com a chamada MQINQ.

### ***TCPChannels (MQLONG)***

Este é o número máximo de canais que podem ser atuais ou clientes que podem ser conectados que usam o protocolo de transmissão TCP/IP.

O valor deve estar no intervalo de 0 a 9999, com um valor padrão de 200. Se você especificar 0, TCP/IP não será utilizado.

Esse atributo é suportado apenas no z/OS

Para determinar o valor desse atributo, use o seletor MQIA\_TCP\_CHANNELS com a chamada MQINQ.

### ***TCPKeepAlive (MQLONG)***

Isso especifica se deve usar TCP KEEPALIVE para verificar se a outra extremidade da conexão ainda está disponível. Se ela não estiver disponível, o canal será fechado.

O valor é um dos seguintes:

#### **MQTCPKEEP\_YES**

Use o TCP KEEPALIVE conforme especificado no conjunto de dados de configuração do perfil TCP Se você especificar o atributo do canal KeepAliveInterval (KAINT), o valor para o qual ele é configurado será usado..

#### **MQTCPKEEP\_NO**

Não utilize TCP KEEPALIVE. Esse é o valor-padrão.

Esse atributo é suportado apenas no z/OS

Para determinar o valor desse atributo, use o seletor MQIA\_TCP\_KEEP\_ALIVE com a chamada MQINQ..

### ***TCPName (MQCHAR8)***

Este é o nome da pilha TCP/IP única ou preferencial que será usada, dependendo do valor de TCPStackType. Esse parâmetro é aplicável apenas em ambientes de várias pilhas do CINET O valor padrão é TCPIP.

Esse atributo é suportado apenas no z/OS

Para determinar o valor desse atributo, use o seletor MQCA\_TCP\_NAME com a chamada MQINQ.. O comprimento deste atributo é fornecido por MQ\_TCP\_NAME\_LENGTH

### ***TCPStackType (MQLONG).***

Isso especifica se o inicializador de canais pode usar apenas a pilha TCP/IP especificada no TCPName ou, opcionalmente, ligar a qualquer pilha TCP/IP selecionada. Esse parâmetro é aplicável apenas em ambientes de várias pilhas do CINET

O valor é um dos seguintes:

#### **MQTCPSTACK\_SINGLE**

O inicializador de canais pode usar apenas os espaços de endereço TCP/IP nomeados em TCPName... Esse é o valor-padrão.

#### **MQTCPSTACK\_MULTIPLE**

O inicializador de canais pode usar qualquer espaço de endereço TCP/IP disponível para ele. Ele será padronizado para aquele especificado em TCPName se nenhum outro for especificado para um canal ou listener..

Esse atributo é suportado apenas no z/OS

Para determinar o valor desse atributo, use o seletor MQIA\_TCP\_STACK\_TYPE com a chamada MQINQ.

### ***TraceRouteGravação (MQLONG)***

Isso controla a gravação de informações de rastreamento de rotas

O valor é um dos seguintes:

#### **MQRECORDING\_DISABLED**

Não é permitido anexar a mensagens de rastreamento de rotas.

#### **MQRECORDING\_Q**

Coloque as mensagens de rastreamento de rotas na fila nomeada fixa

#### **MQRECORDING\_MSG**

Coloque as mensagens de rastreamento de rotas em uma fila determinada usando a própria mensagem  
Este é o valor padrão

Para determinar o valor desse atributo, use o seletor MQIA\_TRACE\_ROUTE\_RECORDING com a chamada MQINQ

### ***TriggerInterval (MQLONG)***

Esse é um intervalo de tempo (em milissegundos) usado para restringir o número de mensagens do acionador.. Isso é relevante apenas quando o *TriggerType* é MQTT\_FIRST Nesse caso, as mensagens do acionador geralmente são geradas apenas quando uma mensagem adequada chega na fila e a fila estava anteriormente vazia Sob determinadas circunstâncias, no entanto, uma mensagem do acionador adicional pode ser gerada com o acionamento MQTT\_FIRST, mesmo se a fila não estava vazia Essas mensagens do acionador adicionais não são geradas com mais frequência do que a cada *TriggerInterval* milissegundos

Para obter mais informações sobre acionamento, consulte [Canais de acionamento](#).

O valor não é menor que 0 e não maior que 999 999 999. O valor padrão é 999 999 999.

Para determinar o valor desse atributo, use o seletor MQIA\_TRIGGER\_INTERVAL com a chamada MQINQ

### ***TriggerInterval (MQLONG)***

Esse é um intervalo de tempo (em milissegundos) usado para restringir o número de mensagens do acionador.. Isso é relevante apenas quando o *TriggerType* é MQTT\_FIRST Nesse caso, as mensagens do acionador geralmente são geradas apenas quando uma mensagem adequada chega na fila e a fila estava anteriormente vazia Sob determinadas circunstâncias, no entanto, uma mensagem do acionador adicional pode ser gerada com o acionamento MQTT\_FIRST, mesmo se a fila não estava vazia Essas mensagens do acionador adicionais não são geradas com mais frequência do que a cada *TriggerInterval* milissegundos

Para obter mais informações sobre acionamento, consulte [Canais de acionamento](#).

O valor não é menor que 0 e não maior que 999 999 999. O valor padrão é 999 999 999.

Para determinar o valor desse atributo, use o seletor MQIA\_TRIGGER\_INTERVAL com a chamada MQINQ

### **Versão (MQCFST)**

Esta é a versão do código IBM MQ como VVRRMMFF, em que:

VV-Versão

RR-Liberação

MM-Nível de manutenção

FF-Nível de correção

### **XrCapability(MQLONG)**

Isso controla se comandos MQ Telemetry são suportados pelo gerenciador de filas.

O valor é um dos seguintes:

#### **MQCAP\_SUPPORTED**

O componente MQ Telemetry instalado e comandos Telemetry são suportados.

#### **MQCAP\_NOT\_SUPPORTED**

Componente MQ Telemetry não instalado.

Esse atributo é suportado apenas no [Multiplataformas](#)

Para determinar o valor desse atributo, use o seletor MQIA\_XR\_CAPABILITY com a chamada MQINQ

## **Atributos para filas**

Há cinco tipos de definição de fila. Alguns atributos de fila se aplicam a todos os tipos de fila; outros atributos de fila se aplicam apenas a determinados tipos de fila

## **Tipos de fila**

O gerenciador de filas suporta os seguintes tipos de definição de filas:

### **Fila local**

É possível armazenar mensagens em uma fila local..

**z/OS** No z/OS , é possível torná-lo uma fila compartilhada ou privada

Uma fila é conhecida em um programa como *local* se for de propriedade do gerenciador de filas ao qual o programa está conectado. É possível obter mensagens de filas locais e colocar as mensagens nelas.

O objeto de definição de fila contém as informações de definição da fila, bem como as mensagens físicas colocadas na fila.

### **Fila do gerenciador de filas locais**

A fila existe no gerenciador de fila local.

**z/OS** A fila é conhecida como uma fila privada no z/OS

### **z/OS Fila compartilhada (z/OS apenas)**

A fila existe em um repositório compartilhado que é acessível a todos os gerenciadores de filas que pertencem ao grupo de filas compartilhadas que possui o repositório compartilhado

Aplicativos conectados a qualquer gerenciador de filas no grupo de filas compartilhadas podem colocar mensagens e remover mensagens de filas desse tipo. Essas filas são efetivamente iguais às filas locais. O valor do atributo da fila **QType** é MQQT\_LOCAL.

Aplicativos conectados ao gerenciador de filas locais podem colocar mensagens e remover mensagens de filas desse tipo. O valor do atributo da fila **QType** é MQQT\_LOCAL.

## Fila de cluster

É possível armazenar mensagens em uma fila de clusters no gerenciador de filas no qual ela está definida. Uma fila de clusters é uma fila que é hospedada por um gerenciador de filas do cluster e disponibilizada para outros gerenciadores de filas no cluster. O valor do atributo da fila **QType** é `MQQT_CLUSTER`.

Uma definição de fila de clusters é divulgada para outros gerenciadores de filas no cluster. Os outros gerenciadores de filas no cluster podem colocar mensagens em uma fila de clusters sem a necessidade de uma definição de fila remota correspondente. Uma fila de clusters pode ser divulgada em mais de um cluster usando uma lista de nomes de cluster.

Quando uma fila é divulgada, qualquer gerenciador de filas no cluster pode colocar mensagens nela. Para colocar uma mensagem, o gerenciador de filas deve descobrir, a partir dos repositórios completos, onde a fila está hospedada. Em seguida, ele inclui algumas informações de roteamento na mensagem e coloca a mensagem em uma fila de transmissão do cluster.

Um gerenciador de filas pode armazenar mensagens para outros gerenciadores de filas em um cluster em várias filas de transmissão. É possível configurar um gerenciador de filas para armazenar mensagens em várias filas de transmissão do cluster de duas maneiras diferentes. Ao configurar o atributo gerenciador de filas **DEFCLXQ** para `CHANNEL`, uma fila de transmissão de cluster diferente é criada automaticamente a partir de `SYSTEM.CLUSTER.TRANSMIT.MODEL.QUEUE` para cada canal de emissor de cluster. Se você configurar a opção da fila de transmissão `CLCHNAME` para corresponder a um ou mais canais do emissor de clusters, o gerenciador de filas pode armazenar mensagens para os canais correspondentes nessa fila de transmissão.



**Atenção:** Se você estiver usando `SYSTEM.CLUSTER.TRANSMIT.QUEUES` dedicado com um gerenciador de filas atualizado a partir de uma versão do produto anterior a IBM WebSphere MQ 7.5, certifique-se de que o `SYSTEM.CLUSTER.TRANSMIT.MODEL.QUEUE` tenha a opção `SHARE/NOSHARE` definida como **SHARE**.



Uma fila de clusters pode ser uma fila que é compartilhada por membros de um grupo de filas compartilhadas no IBM MQ for z/OS.

## Fila remota

Uma fila remota não é uma fila física; é a definição local de uma fila que existe em um gerenciador de filas remotas. A definição local da fila remota contém informações que informam ao gerenciador de filas locais como rotear mensagens para o gerenciador de filas remotas.

Os aplicativos conectados ao gerenciador de filas locais podem colocar mensagens em filas desse tipo; as mensagens são colocadas na fila de transmissão local usada para rotear mensagens para o gerenciador de filas remotas. Aplicativos não podem remover mensagens de filas remotas. O valor do atributo da fila **QType** é `MQQT_REMOTE`.

Também é possível utilizar uma definição de fila remota para:

- Alias da fila de respostas

Nesse caso, o nome da definição é o nome de uma fila de resposta.. Para obter mais informações, consulte [Aliases e clusters da fila de resposta](#)

- Alias do gerenciador de filas

Neste caso, o nome da definição é um alias para um gerenciador de filas e não o nome de uma fila. Para obter mais informações, consulte [Aliases e clusters do gerenciador de filas](#).

## Fila de alias

Esta não é uma fila física; ela é um nome alternativo para uma fila local, uma fila compartilhada, uma fila de clusters ou uma fila remota. O nome da fila para a qual o alias é resolvido faz parte da definição da fila de alias.

Os aplicativos conectados ao gerenciador de filas locais podem colocar mensagens em filas desse tipo; as mensagens são colocadas na fila para a qual o alias é resolvido. Os aplicativos podem remover mensagens de filas desse tipo se o alias for resolvido para uma fila local, uma fila compartilhada, ou uma fila de clusters que possui uma instância local. O valor do atributo da fila **QType** é `MQQT_ALIAS`.

## Fila modelo

Esta não é uma fila física; ela é um conjunto de atributos de fila a partir do qual uma fila local pode ser criada

As mensagens não podem ser armazenadas em filas deste tipo

## limites da fila

No IBM MQ 9.2.0, você tem a opção de configurar e monitorar filas que suportarão substancialmente mais do que o limite padrão de dois terabytes usado em liberações anteriores do IBM MQ. Você também tem a opção de reduzir o tamanho que um arquivo de fila pode crescer.

Para permitir que você configure filas, é possível usar o atributo **MAXFSIZE** em filas locais e de modelo e para monitorar filas, é possível usar os atributos de status da fila **CURFSIZE** e **CURMAXFS**.

Para obter mais informações, consulte [Modificando arquivos de fila do IBM MQ](#).

## Atributos da Fila

Alguns atributos de fila se aplicam a todos os tipos de fila; outros atributos de fila se aplicam apenas a determinados tipos de fila. Os tipos de fila aos quais um atributo se aplica são mostrados em [Tabela 561 na página 867](#) e tabelas subsequentes..

[Tabela 561 na página 867](#) resume os atributos específicos para filas. Os atributos são descritos em ordem alfabética

**Nota:** Os nomes dos atributos mostrados nesta seção são nomes descritivos usados com as chamadas MQINQ e MQSET ; os nomes são os mesmos dos comandos PCF. Quando comandos MQSC são usados para definir, alterar ou exibir atributos, nomes abreviados alternativos são usados; consulte [Comandos MQSC](#) para obter detalhes.

Na tabela a seguir, as colunas se aplicam da seguinte forma:

- A coluna para filas locais também se aplica a filas compartilhadas..
- A coluna para filas modelo indica quais atributos são herdados pela fila local criada da fila modelo.
- A coluna para filas de clusters indica os atributos que podem ser consultados quando a fila de clusters é aberta para consulta sozinha, ou para consulta e saída. Se quaisquer outros atributos forem consultados, a chamada retorna o código de conclusão MQCC\_WARNING e o código de razão MQRC\_SELECTOR\_NOT\_FOR\_TYPE (2068).

Se a fila de clusters for aberta para consulta mais uma ou mais de entrada, navegação ou configuração, a coluna para filas locais se aplicará em vez disso..

Se a fila de clusters for aberta para consulta sozinha, ou para consulta e saída, além de especificar o nome do gerenciador de filas base, a coluna para filas locais será aplicada em seu lugar.

Atributo	Descrição	Local	Modelo	Alias	Remoto	Cluster
<a href="#">AlterationDate</a>	Data em que a definição foi alterada pela última vez	X		X	X	
<a href="#">AlterationTime</a>	Horário em que a definição foi alterada pela última vez	X		X	X	
<a href="#">BackoutRequeueQName</a>	Nome excessivo da fila de reenqueueamento de restauração	X	X			
<a href="#">BackoutThreshold</a>	Limite de Recuperação	X	X			
<a href="#">BaseQName</a>	Nome da fila para o qual o alias é resolvido			X		
<a href="#">CFStrucName</a>	Nome da estrutura do recurso de acoplamento	X	X			

Tabela 561. Atributos para filas (continuação)

Atributo	Descrição	Local	Modelo	Alias	Remoto	Cluster
<u>CLCHNAME</u>	Nomes do canal do emissor de clusters	✓	✓			
<u>ClusterName</u>	Nome do cluster ao qual a fila pertence	X		X	X	X
<u>ClusterNameList</u>	Nome do objeto da lista de nomes contendo nomes de clusters aos quais a fila pertence	X		X	X	
<u>CLWLQueuePriority</u>	prioridade da fila de cargas de trabalho do cluster	X		X	X	X
<u>CLWLQueueRank</u>	Classificação da fila de cargas de trabalho do cluster	X		X	X	X
<u>CLWLUseQ</u>	Usar fila remota	X				
<u>CreationDate</u>	Data em que a fila foi criada	X				
<u>CreationTime</u>	Horário em que a fila foi criada	X				
<u>CurrentQDepth</u>	Profundidade de fila atual	X				
<u>DefaultPutResponse</u>	Resposta Put Padrão	✓	✓	✓	✓	
<u>DefBind</u>	Ligação padrão	X		X	X	X
<u>DefinitionType attribute</u>	Tipo de definição de fila	X	X			
<u>DefInputOpenOption</u>	Opções de abertura de entrada padrão	X	X			
<u>DefPersistence</u>	Persistência padrão de mensagem	X	X	X	X	X
<u>DefPriority</u>	Prioridade padrão da mensagem	✓	✓	✓	✓	✓
<u>DefReadAhead</u>	Padrão lido antes	X	X	X		
<u>DistLists</u>	Suporte à lista de distribuição	X	X			
<u>HardenGetBackout</u>	Se deve-se manter uma contagem de restauração precisa	X	X			
<u>IndexType</u>	Tipo de índice	X	X			
<u>InhibitGet</u>	Se as operações get para a fila são permitidas	X	X	X		
<u>InhibitPut</u>	Se as operações put para a fila são permitidas	X	X	X	X	X
<u>InitiationQName</u>	Nome da fila de inicialização	X	X			
<u>MaxMsgLength</u>	Tamanho máximo da mensagem em bytes	X	X			
<u>MaxQDepth</u>	Profundidade máxima da fila	X	X			
<u>MsgDeliverySequence attribute</u>	Seqüência de entrega de mensagens	X	X			
<u>NonPersistentMessage Class</u>	Objetivo de confiabilidade para mensagens não persistentes	X	X			
<u>OpenInputCount</u>	Número de aberturas para entrada	X				
<u>OpenOutputCount</u>	Número de aberturas para saída	X				
<u>PropertyControl</u>	Controle de propriedade	✓	✓	✓		
<u>ProcessName</u>	Nome do processo	X	X			



*Tabela 561. Atributos para filas (continuação)*

<b>Atributo</b>	<b>Descrição</b>	<b>Local</b>	<b>Modelo</b>	<b>Alias</b>	<b>Remoto</b>	<b>Cluster</b>
<a href="#">QDepthHighEvent attribute</a>	Se eventos de Alta Profundidade da Fila são gerados	X	X			
<a href="#">QDepthHighLimit</a>	Limite alto para profundidade da fila	X	X			
<a href="#">QDepthLowEvent attribute</a>	Se os eventos de Baixa Profundidade da Fila são gerados	X	X			
<a href="#">QDepthLowLimit attribute</a>	Limite baixo para profundidade da fila	X	X			
<a href="#">QDepthMaxEvent</a>	Se eventos de Fila Cheia são gerados	X	X			
<a href="#">QDesc</a>	Descrição da fila	X	X	X	X	X
<a href="#">QName</a>	Nome da fila	X		X	X	X
<a href="#">QServiceInterval</a>	Destino para intervalo de serviço de fila	X	X			
<a href="#">QServiceIntervalEvent attribute</a>	Se os eventos de Intervalo de Serviço Alto ou Intervalo de Serviço OK são gerados	X	X			
<a href="#">QSGDisp attribute</a>	Disposição do Grupo de Compartilhamento de Fila	X		X	X	
<a href="#">QueueAccounting</a>	Coleta de dados de contabilidade da fila	X	X	X	X	X
<a href="#">QueueMonitoring</a>	Dados de monitoramento on-line para filas	X	✓			
<a href="#">QueueStatistics</a>	coleta de dados de estatísticas de fila	X	X	X	X	X
<a href="#">QType</a>	Tipo de fila	X		X	X	X
<a href="#">RemoteQMgrName</a>	Nome do gerenciador de fila remoto				X	
<a href="#">RemoteQName</a>	Nome da fila remota				X	
<a href="#">RetentionInterval</a>	Intervalo de retenção	X	X			
<a href="#">Scope</a>	Se uma entrada para a fila também existe em um diretório de célula	X		X	X	
<a href="#">Shareability</a>	Compartilhamento de fila	X	X			
<a href="#">StorageClass</a>	Classe de armazenamento para fila	X	X			
<a href="#">TriggerControl</a>	Controle de ativação	X	X			
<a href="#">TriggerData</a>	Dados de Ativação	X	X			
<a href="#">TriggerDepth</a>	Profundidade de ativação	X	X			
<a href="#">TriggerMsgPriority</a>	Prioridade da mensagem limite para acionadores	X	X			
<a href="#">TriggerType</a>	Tipo de ativação	X	X			
<a href="#">Usage attribute</a>	Uso da Fila	X	X			
<a href="#">XmitQName</a>	Nome da fila de transmissão				X	

### **Conceitos relacionados**

[Filas de Clusters](#)

[Filas locais](#)

[Como escolher o tipo de fila de transmissão do cluster a usar](#)

### ***AlterationDate (MQCHAR12)***

Data da última alteração da definição.

Tabela 562. Tipos de fila aos quais este atributo se aplica				
Local	Modelo	Alias	Remoto	Cluster
X		X	X	

Esta é a data em que a definição foi mudada pela última vez. O formato da data é YYYY-MM-DD, preenchido com dois espaços em branco finais para tornar o comprimento de 12 bytes (por exemplo, 1992-09-23--), em que -- representa dois caracteres em branco).

Os valores de determinados atributos (por exemplo, *CurrentQDepth*) mudam conforme o gerenciador de filas opera. As alterações nesses atributos não afetam *AlterationDate*.

Para determinar o valor desse atributo, use o seletor MQCA\_ALTERATION\_DATE com a chamada MQINQ. O comprimento desse atributo é fornecido por MQ\_DATE\_LENGTH.

### **AlterationTime (MQCHAR8)**

Horário em que a definição foi mudada pela última vez.

Tabela 563. Tipos de fila aos quais este atributo se aplica				
Local	Modelo	Alias	Remoto	Cluster
X		X	X	

Este é o horário em que a definição foi alterada pela última vez. O formato do horário é HH.MM.SS usando o relógio de 24 horas, com um zero à esquerda se a hora for menor que 10 (por exemplo 09.10.20).

- No z/OS, o horário é Horário de Greenwich (GMT), sujeito ao relógio do sistema ser configurado com precisão para GMT.
- Em outros ambientes, o horário é local.

Os valores de determinados atributos (por exemplo, *CurrentQDepth*) mudam conforme o gerenciador de filas opera. As alterações nesses atributos não afetam *AlterationTime*.

Para determinar o valor desse atributo, use o seletor MQCA\_ALTERATION\_TIME com a chamada MQINQ.. O comprimento desse atributo é fornecido por MQ\_TIME\_LENGTH.

### **BackoutRequeueQName (MQCHAR48)**

Este é o nome da fila de reenfileiramento de restauração excessiva. Além de permitir que seu valor seja consultado, o gerenciador de filas não executa nenhuma ação baseada no valor desse atributo.

Tabela 564. Tipos de fila aos quais este atributo se aplica				
Local	Modelo	Alias	Remoto	Cluster
X	X			

Os aplicativos em execução no WebSphere Application Server e aqueles que usam o IBM MQ Application Server Facilities usam esse atributo para determinar para onde as mensagens que foram restauradas devem ir. Para todos os outros aplicativos, o gerenciador de filas não executa nenhuma ação baseada no valor do atributo.

O IBM MQ classes for JMS usa esse atributo para determinar onde transferir uma mensagem que já foi restaurada o número máximo de vezes, conforme especificado pelo atributo *BackoutThreshold*.

Para determinar o valor desse atributo, use o seletor MQCA\_BACKOUT\_REQ\_Q\_NAME com a chamada MQINQ.. O comprimento desse atributo é fornecido por MQ\_Q\_NAME\_LENGTH.

### **BackoutThreshold (MQLONG)**

Este é o limite de restauração.. Além de permitir que seu valor seja consultado, o gerenciador de filas não executa nenhuma ação baseada no valor desse atributo.

<i>Tabela 565. Tipos de fila aos quais este atributo se aplica</i>				
Local	Modelo	Alias	Remoto	Cluster
X	X			

Aplicativos em execução dentro do WebSphere Application Server e aqueles que usam o IBM MQ Application Server Facilities usarão esse atributo para determinar se uma mensagem deve ser restaurada. Para todos os outros aplicativos, o gerenciador de filas não executa nenhuma ação baseada no valor do atributo.

IBM MQ classes for JMS usa esse atributo para determinar quantas vezes permitir que uma mensagem seja restaurada antes de transferir a mensagem para a fila especificada pelo atributo *BackoutRequeueQName* ..

Para determinar o valor desse atributo, use o seletor MQIA\_BACKOUT\_THRESHOLD com a chamada MQINQ

### **BaseQName (MQCHAR48)**

Este é o nome de uma fila definida para o gerenciador de fila local.

<i>Tabela 566. Tipos de fila aos quais este atributo se aplica</i>				
Local	Modelo	Alias	Remoto	Cluster
		X		

(Para obter mais informações sobre nomes de filas, consulte [MQOD-campo ObjectName.](#)) A fila é um dos seguintes tipos:

#### **MQQT\_LOCAL**

Fila local.

#### **MQQT\_REMOTE**

A definição local de uma fila remota

#### **MQQT\_CLUSTER**

Fila de clusters

Para determinar o valor desse atributo, use o seletor MQCA\_BASE\_Q\_NAME com a chamada MQINQ O comprimento desse atributo é fornecido por MQ\_Q\_NAME\_LENGTH

### **BaseType (MQCFIN)**

O tipo de objeto para o qual o alias é resolvido

<i>Tabela 567. Tipos de fila aos quais este atributo se aplica</i>				
Local	Modelo	Alias	Remoto	Cluster
		X		

Ele é um dos seguintes valores:

#### **MQOT\_Q**

O tipo de objeto base é uma fila

#### **MQOT\_TOPIC**

O tipo de objeto base é um tópico

### **CFStrucName (MQCHAR12)**

Esse é o nome da estrutura do recurso de acoplamento em que as mensagens na fila são armazenadas. O primeiro caractere do nome está no intervalo de A a Z e os caracteres restantes estão no intervalo de A a Z, 0 a 9 ou em branco.

Tabela 568. Tipos de fila aos quais este atributo se aplica				
Local	Modelo	Alias	Remoto	Cluster
X	X			

Para obter o nome completo da estrutura no recurso de acoplamento, sufixe o valor do atributo do gerenciador de fila **QSGName** com o valor do atributo da fila **CFStrucName**.

Esse atributo se aplica apenas a filas compartilhadas; ele será ignorado se *QSGDisp* não tiver o valor **MQQSGD\_SHARED**.

Para determinar o valor desse atributo, use o seletor **MQCA\_CF\_STRUC\_NAME** com a chamada **MQINQ**. O comprimento deste atributo é fornecido por **MQ\_CF\_STRUC\_NAME\_LENGTH**.

 Esse atributo é suportado apenas no z/OS

### **ClusterChannelName (MQCHAR20)**

**ClusterChannelName** é o nome genérico dos canais do emissor de clusters que usam essa fila como uma fila de transmissão. O atributo especifica quais canais de emissor de clusters enviam mensagens a um canal do receptor de clusters a partir desta fila de transmissão do cluster.

Tabela 569. Tipos de fila aos quais este atributo se aplica				
Local	Modelo	Alias	Remoto	Cluster
X	X			

A configuração padrão do gerenciador de filas é para todos os canais do emissor de clusters para enviar mensagens a partir de uma única fila de transmissão, **SYSTEM.CLUSTER.TRANSMIT.QUEUE**. Muda-se a configuração padrão modificando o atributo gerenciador de filas, **DefClusterXmitQueueType**. O valor padrão do atributo é **SCTQ**. É possível alterar o valor para **CHANNEL**. Ao configurar o atributo **DefClusterXmitQueueType** para **CHANNEL**, cada canal do emissor de cluster será padronizado para usar uma fila específica de transmissões de clusters, **SYSTEM.CLUSTER.TRANSMIT.ChannelName**.

É possível também configurar o atributo de fila de transmissão **ClusterChannelName** para um canal do emissor de clusters manualmente. As mensagens que são destinadas ao gerenciador de filas conectado pelo canal do emissor de clusters são armazenadas na fila de transmissão que identifica o canal do emissor de clusters. Elas não são armazenadas na fila de transmissão do cluster padrão. Se você configurar o atributo **ClusterChannelName** com espaços em branco, o canal alternará para a fila de transmissão do cluster padrão quando o canal for reiniciado. A fila padrão é de **SYSTEM.CLUSTER.TRANSMIT.ChannelName** ou **SYSTEM.CLUSTER.TRANSMIT.QUEUE**, dependendo do valor do atributo gerenciador de filas **DefClusterXmitQueueType**.

Ao especificar asteriscos, "\*", em **ClusterChannelName**, é possível associar uma fila de transmissão a um conjunto de canais de emissor de clusters. Os asteriscos podem estar no início, fim ou qualquer número de locais no meio da sequência do nome de canal. **ClusterChannelName** é limitado a 20 caracteres de comprimento: **MQ\_CHANNEL\_NAME\_LENGTH**.

### **ClusterName (MQCHAR48)**

É o nome do cluster ao qual a fila pertence.

Tabela 570. Tipos de fila aos quais este atributo se aplica				
Local	Modelo	Alias	Remoto	Cluster
X		X	X	X

Se a fila pertencer a mais de um cluster, *ClusterNameList* especifica o nome de um objeto de lista de nomes que identifica os clusters e *ClusterName* está em branco. Pelo menos um de *ClusterName* e *ClusterNameList* deve estar em branco

Para determinar o valor desse atributo, use o seletor MQCA\_CLUSTER\_NAME com a chamada MQINQ. O comprimento desse atributo é fornecido por MQ\_CLUSTER\_NAME\_LENGTH

### **ClusterNameList (MQCHAR48)**

Esse é o nome de um objeto de lista de nomes que contém os nomes de clusters aos quais essa fila pertence

<i>Tabela 571. Tipos de fila aos quais este atributo se aplica</i>				
Local	Modelo	Alias	Remoto	Cluster
X		X	X	

Se a fila pertencer a apenas um cluster, o objeto namelist contém apenas um nome. Como alternativa, *ClusterName* pode ser usado para especificar o nome do cluster, em cujo caso, *ClusterNameList* está em branco. Pelo menos um de *ClusterName* e *ClusterNameList* deve estar em branco

Para determinar o valor desse atributo, use o seletor MQCA\_CLUSTER\_NAMELIST com a chamada MQINQ. O comprimento desse atributo é fornecido por MQ\_NAMELIST\_NAME\_LENGTH

### **CLWLQueuePriority (MQLONG)**

Esta é a prioridade da fila de carga de trabalho do cluster, um valor no intervalo de 0 a 9 que representa a prioridade da fila

<i>Tabela 572. Tipos de fila aos quais este atributo se aplica</i>				
Local	Modelo	Alias	Remoto	Cluster
X		X	X	X

Para obter mais informações, consulte [Filas de clusters](#)

Para determinar o valor desse atributo, use o seletor MQIA\_CLWL\_Q\_PRIORITY com a chamada MQINQ

### **CLWLQueueRank (MQLONG)**

Esta é a classificação da fila de cargas de trabalho do cluster, um valor no intervalo de 0 a 9 que representa a classificação da fila

<i>Tabela 573. Tipos de fila aos quais este atributo se aplica</i>				
Local	Modelo	Alias	Remoto	Cluster
X		X	X	X

Para obter mais informações, consulte [Filas de clusters](#)

Para determinar o valor desse atributo, use o seletor MQIA\_CLWL\_Q\_RANK com a chamada MQINQ.

### **CLWLUseQ (MQLONG)**

Isso define o comportamento de um MQPUT quando a fila de destino tem uma instância local e pelo menos uma instância de cluster remota. Se o envio originar-se de um canal de cluster, esse atributo não será aplicável.

<i>Tabela 574. Tipos de fila aos quais este atributo se aplica</i>				
Local	Modelo	Alias	Remoto	Cluster
X				

O valor é um dos seguintes:

**MQCLWL\_USEQ\_ANY**

Use filas remotas e locais..

**MQCLWL\_USEQ\_LOCAL**

Não use filas remotas..

**MQCLWL\_USEQ\_AS\_Q\_MGR**

Herde a definição do MQIA\_CLWL\_USEQ do gerenciador de fila

Para obter mais informações, consulte [Filas de clusters](#)

Para determinar o valor desse atributo, use o seletor MQIA\_CLWL\_USEQ com a chamada MQINQ. O comprimento desse atributo é fornecido por MQ\_CLWL\_USEQ\_LENGTH

**CreationDate (MQCHAR12)**

Essa é a data em que a fila foi criada.

<i>Tabela 575. Tipos de fila aos quais este atributo se aplica</i>				
Local	Modelo	Alias	Remoto	Cluster
X				

O formato da data é YYYY-MM-DD, preenchido com dois espaços em branco finais para criar o comprimento de 12 bytes (por exemplo, 2013-09-23-- , em que -- representa 2 caracteres em branco).

- No IBM i, a data de criação de uma fila pode ser diferente da data da entidade do sistema operacional subjacente (arquivo ou espaço do usuário) que representa a fila

Para determinar o valor desse atributo, use o seletor MQCA\_CREATION\_DATE com a chamada MQINQ.. O comprimento desse atributo é fornecido por MQ\_CREATION\_DATE\_LENGTH

**CreationTime (MQCHAR8)**

Este é o horário em que a fila foi criada

<i>Tabela 576. Tipos de fila aos quais este atributo se aplica</i>				
Local	Modelo	Alias	Remoto	Cluster
X				

O formato do horário é HH.MM.SS usando o relógio de 24 horas, com um zero à esquerda se a hora for menor que 10 (por exemplo 09.10.20).

- No z/OS, o horário é Horário de Greenwich (GMT), sujeito ao relógio do sistema ser configurado com precisão para GMT.
- Em outros ambientes, o horário é local.
- No IBM i, o tempo de criação de uma fila pode diferir daquele da entidade do sistema operacional subjacente (arquivo ou espaço do usuário) que representa a fila...

Para determinar o valor desse atributo, use o seletor MQCA\_CREATION\_TIME com a chamada MQINQ.. O comprimento desse atributo é fornecido por MQ\_CREATION\_TIME\_LENGTH

**CurrentQDepth (MQLONG)**

Esse é o número de mensagens atualmente na fila.

<i>Tabela 577. Tipos de fila aos quais este atributo se aplica</i>				
Local	Modelo	Alias	Remoto	Cluster
X				

Ele é incrementado durante uma chamada MQPUT e durante a restauração de uma chamada MQGET. Ele é decrementado durante uma chamada MQGET não de navegação e durante a restauração de uma chamada MQPUT. O efeito disso é que a contagem inclui mensagens que foram colocadas na fila dentro de uma unidade de trabalho, mas que ainda não foram confirmados, mesmo que eles não sejam elegíveis para serem recuperadas pela chamada MQGET. Da mesma forma, ele exclui mensagens que foram recuperadas dentro de uma unidade de trabalho usando a chamada MQGET, mas que ainda precisam ser confirmados.

A contagem também inclui mensagens que passaram de seu tempo de expiração, mas ainda não foram descartados, embora essas mensagens não sejam elegíveis para serem recuperados. Consulte MQMD-Campo de expiração para obter mais informações..

O processamento da unidade de trabalho e a segmentação de mensagens podem fazer com que o *CurrentQDepth* exceda *MaxQDepth*. No entanto, isso não afeta a capacidade de recuperação das mensagens; todas as mensagens na fila podem ser recuperadas usando a chamada MQGET da maneira normal.

O valor desse atributo flutua conforme o gerenciador de filas opera.

Para determinar o valor desse atributo, use o seletor MQIA\_CURRENT\_Q\_DEPTH com a chamada MQINQ..

### **Resposta de DefaultPut(MQLONG)**

Especifica o tipo de resposta a ser usado para operações put para a fila quando um aplicativo especifica MQPMO\_RESPONSE\_AS\_Q\_DEF.

*Tabela 578. Tipos de fila aos quais este atributo se aplica*

Local	Modelo	Alias	Remoto	Cluster
X	X	X	X	

Ele é um dos seguintes valores:

#### **MQPRT\_SYNC\_RESPONSE**

A operação put é emitida de forma síncrona, retornando uma resposta..

#### **MQPRT\_ASYNC\_RESPONSE**

A operação put é emitida assincronamente, retornando um subconjunto de campos MQMD.

### **DefBind (MQLONG)**

Essa é a ligação padrão usada quando MQOO\_BIND\_AS\_Q\_DEF é especificado na chamada MQOPEN e a fila é uma fila de clusters.

*Tabela 579. Tipos de fila aos quais este atributo se aplica*

Local	Modelo	Alias	Remoto	Cluster
X		X	X	X

O valor é um dos seguintes:

#### **MQBND\_BIND\_ON\_OPEN**

Ligação fixa pela chamada MQOPEN.

#### **MQBND\_BIND\_NOT\_FIXED**

Ligação não corrigida.

#### **MQBND\_BIND\_ON\_GROUP**

Permite que um aplicativo solicite que um grupo de mensagens seja alocado na mesma instância de destino.

Para determinar o valor desse atributo, use o seletor MQIA\_DEF\_BIND com a chamada MQINQ.

### **DefinitionType (MQLONG)**

Isso indica como a fila foi definida.

Tabela 580. Tipos de fila aos quais este atributo se aplica				
Local	Modelo	Alias	Remoto	Cluster
X	X			

O valor é um dos seguintes:

#### **MQODT\_PREDEFINED**

A fila é uma fila permanente criada pelo administrador do sistema; somente ele pode excluí-la.

As filas predefinidas são criadas usando o comando MQSC DEFINE e podem ser excluídas apenas usando o comando MQSC DELETE . Filas predefinidas não podem ser criadas a partir de filas modelo.

Comandos podem ser emitidos por um operador ou por um usuário autorizado enviando uma mensagem de comando para a fila de entrada de comando (consulte [CommandInputatributo QName](#) para obter mais informações).

#### **MQODT\_PERMANENT\_DYNAMIC**

A fila é uma fila permanente que foi criada por um aplicativo emitindo uma chamada MQOPEN com o nome de uma fila modelo especificada no descritor de objeto MQOD A definição de fila modelo tinha o valor MQODT\_PERMANENT\_DYNAMIC para o atributo **DefinitionType** .

Esse tipo de fila pode ser excluído usando a chamada MQCLOSE Consulte [“MQCLOSE-Fechar objeto” na página 673](#) para obter mais detalhes.

O valor do atributo **QSGDisp** para uma fila dinâmica permanente é MQQSGD\_Q\_MGR.

#### **MQODT\_TEMPORARY\_DYNAMIC**

A fila é uma fila temporária que foi criada por um aplicativo emitindo uma chamada MQOPEN com o nome de uma fila modelo especificada no descritor de objeto MQOD.. A definição de fila modelo tinha o valor MQODT\_TEMPORARY\_DYNAMIC para o atributo **DefinitionType** .

Esse tipo de fila é excluído automaticamente pela chamada MQCLOSE quando ele é fechado pelo aplicativo que o criou

O valor do atributo **QSGDisp** para uma fila dinâmica temporária é MQQSGD\_Q\_MGR.

#### **MQODT\_SHARED\_DYNAMIC**

A fila é uma fila permanente compartilhada que foi criada por um aplicativo emitindo uma chamada MQOPEN com o nome de uma fila modelo especificada no MQOD do descritor de objeto A definição de fila modelo tinha o valor MQODT\_SHARED\_DYNAMIC para o atributo **DefinitionType** .

Esse tipo de fila pode ser excluído usando a chamada MQCLOSE Consulte [“MQCLOSE-Fechar objeto” na página 673](#) para obter mais detalhes.

O valor do atributo **QSGDisp** para uma fila dinâmica compartilhada é MQQSGD\_SHARED.

Este atributo em uma definição de fila modelo não indica como a fila modelo foi definida, porque as filas modelo são sempre predefinidas Em vez disso, o valor desse atributo na fila modelo é usado para determinar o *DefinitionType* de cada uma das filas dinâmicas criadas a partir da definição de fila modelo usando a chamada MQOPEN.

Para determinar o valor desse atributo, use o seletor MQIA\_DEFINITION\_TYPE com a chamada MQINQ.

#### **DefInputOpenOption (MQLONG)**

Essa é a maneira padrão na qual abrir a fila para entrada

Tabela 581. Tipos de fila aos quais este atributo se aplica				
Local	Modelo	Alias	Remoto	Cluster
X	X			

Ele se aplicará se a opção MQOO\_INPUT\_AS\_Q\_DEF for especificado na chamada MQOPEN quando a fila for aberta O valor é um dos seguintes:



## **MQOO\_INPUT\_EXCLUSIVE**

Abra a fila para obter mensagens com acesso exclusivo

A fila é aberta para uso com as chamadas MQGET subsequentes. A chamada falha com o código de razão MQRC\_OBJECT\_IN\_USE se a fila estiver atualmente aberta por este ou outro aplicativo para entrada de qualquer tipo (MQOO\_INPUT\_SHARED ou MQOO\_INPUT\_EXCLUSIVE).

## **MQOO\_INPUT\_SHARED**

Abrir fila para obter mensagens com acesso compartilhado.

A fila é aberta para uso com as chamadas MQGET subsequentes. A chamada poderá ser bem-sucedida se a fila estiver atualmente aberta por este ou outro aplicativo com MQOO\_INPUT\_SHARED, mas falhará com o código de razão MQRC\_OBJECT\_IN\_USE se a fila estiver atualmente aberta com MQOO\_INPUT\_EXCLUSIVE.

Para determinar o valor desse atributo, use o seletor MQIA\_DEF\_INPUT\_OPEN\_OPTION com a chamada MQINQ..

## **DefPersistence (MQLONG)**

Esta é a persistência padrão de mensagens na fila. Aplica-se se MQPER\_PERSISTENCE\_AS\_Q\_DEF for especificado no descritor de mensagem quando a mensagem for colocada.

<i>Tabela 582. Tipos de fila aos quais este atributo se aplica</i>				
<b>Local</b>	<b>Modelo</b>	<b>Alias</b>	<b>Remoto</b>	<b>Cluster</b>
X	X	X	X	X

Se houver mais de uma definição no caminho de resolução do nome da fila, a persistência padrão será obtida do valor desse atributo na *primeira* definição no caminho no momento da chamada MQPUT ou MQPUT1. Isto poderia ser:

- uma fila de alias
- Uma fila local
- Uma definição local de uma fila remota
- Um alias do gerenciador de filas
- Uma fila de transmissão (por exemplo, a fila *DefXmitQName*)

O valor é um dos seguintes:

## **MQPER\_PERSISTENT**

A mensagem sobrevive a falhas do sistema e reinicializações do gerenciador de filas. Mensagens persistentes não podem ser colocadas em:

- Filas dinâmicas temporárias
- Filas compartilhadas que mapeiam para um objeto CFSTRUCT em CFLEVEL (2) ou abaixo ou onde o objeto CFSTRUCT é definido como RECOVER (NO).

Mensagens persistentes podem ser colocadas em filas dinâmicas permanentes e filas predefinidas.

## **MQPER\_NOT\_PERSISTENT**

A mensagem normalmente não sobrevive a falhas do sistema ou a reinicializações do gerenciador de filas. Isso se aplicará mesmo se uma cópia intacta da mensagem for localizada no armazenamento auxiliar durante uma reinicialização do gerenciador de filas.

No caso de filas compartilhadas, as mensagens não persistentes *não* sobrevivem a reinicializações de gerenciadores de filas no grupo de filas compartilhadas, mas não sobrevivem a falhas do recurso de acoplamento usado para armazenar mensagens nas filas compartilhadas.

As mensagens persistentes e não persistentes podem existir na mesma fila.

Para determinar o valor desse atributo, use o seletor MQIA\_DEF\_PERSISTENCE com a chamada MQINQ..

### **DefPriority (MQLONG)**

Esta é a prioridade padrão para as mensagens na fila Isso se aplicará se MQPRI\_PRIORITY\_AS\_Q\_DEF for especificado no descritor de mensagem quando a mensagem for colocada na fila

Tabela 583. Tipos de fila aos quais este atributo se aplica				
Local	Modelo	Alias	Remoto	Cluster
X	X	X	X	X

Se houver mais de uma definição no caminho de resolução do nome da fila, a prioridade padrão para a mensagem será obtida do valor desse atributo na *primeira* definição no caminho no momento da operação put. Isto poderia ser:

- uma fila de alias
- Uma fila local
- Uma definição local de uma fila remota
- Um alias do gerenciador de filas
- Uma fila de transmissão (por exemplo, a fila *DefXmitQName* )

A maneira na qual uma mensagem é colocada em uma fila depende do valor do atributo **MsgDeliverySequence** da fila:

- Se o atributo **MsgDeliverySequence** for MQMDS\_PRIORITY, a posição lógica na qual uma mensagem é colocada na fila dependerá do valor do campo *Priority* no descritor da mensagem..
- Se o atributo **MsgDeliverySequence** for MQMDS\_FIFO, as mensagens serão colocadas na fila como se tivessem uma prioridade igual ao *DefPriority* da fila resolvida, independentemente do valor do campo *Priority* no descritor de mensagens.. No entanto, o campo *Priority* retém o valor especificado pelo aplicativo que colocou a mensagem Consulte [MsgDelivery](#) para obter mais informações

As prioridades estão no intervalo de zero (mais baixo) a *MaxPriority* (mais alto); consulte o atributo [MaxPriority](#).

Para determinar o valor desse atributo, use o seletor MQIA\_DEF\_PRIORITY com a chamada MQINQ.

### **DefReadAhead (MQLONG)**

Especifica o comportamento de leitura antecipada padrão para mensagens não persistentes entregues para o cliente

Tabela 584. Tipos de fila aos quais este atributo se aplica				
Local	Modelo	Alias	Remoto	Cluster
X	X	X		

DefReadAhead pode ser configurado para um dos seguintes valores:

#### **MQREADA\_NO**

As mensagens não persistentes não são enviadas antecipadamente para o cliente antes que um aplicativo as solicite No máximo uma mensagem não persistente pode ser perdida se o cliente terminar de forma anormal.

#### **MQREADA\_YES**

As mensagens não persistentes são enviadas antecipadamente para o cliente antes que um aplicativo as solicite As mensagens não persistentes poderão ser perdidas se o cliente for encerrado de forma anormal ou se o cliente não consumir todas as mensagens enviadas.

## **MQREADA\_DISABLED**

Leia antes de mensagens não persistentes em não ativadas para esta fila. As mensagens não são enviadas antecipadamente para o cliente, independentemente de a leitura antecipada ser solicitada pelo aplicativo cliente.

Para determinar o valor desse atributo, use o seletor MQIA\_DEF\_READ\_AHEAD com a chamada MQINQ..

## **DefPResp (MQLONG)**

O atributo default put response type (DEFPRESP) define o valor usado pelos aplicativos quando o Tipo PutResponso no MQPMO foi configurado como MQPMO\_RESPONSE\_AS\_Q\_DEF. Esse atributo é válido para todos os tipos de filas

<i>Tabela 585. Tipos de fila aos quais este atributo se aplica</i>				
<b>Local</b>	<b>Modelo</b>	<b>Alias</b>	<b>Remoto</b>	<b>Cluster</b>
X	X	X	X	X

O valor é um dos seguintes:

### **SÍNC**

A operação put é emitida de forma síncrona retornando uma resposta.

### **ASYN**

A operação put é emitida assincronamente, retornando um subconjunto de campos MQMD.

Para determinar o valor desse atributo, use o seletor MQIA\_DEF\_PUT\_RESPONSE\_TYPE com a chamada MQINQ..

## **DistLists (MQLONG)**

Indica se mensagens da lista de distribuição podem ser colocadas na fila.

<i>Tabela 586. Tipos de fila aos quais este atributo se aplica</i>				
<b>Local</b>	<b>Modelo</b>	<b>Alias</b>	<b>Remoto</b>	<b>Cluster</b>
X	X			

Um agente do canal de mensagens (MCA) configura o atributo para informar ao gerenciador de fila local se o gerenciador de filas na outra extremidade do canal suporta listas de distribuições. Esse último gerenciador de filas (chamado de gerenciador de filas de *parceria*) é aquele que recebe a mensagem em seguida, depois de ter sido removido da fila de transmissão local por um MCA de envio...

O MCA de envio configura o atributo sempre que ele estabelece uma conexão com o MCA de recebimento no gerenciador de filas de parceria Dessa forma, o MCA de envio pode fazer com que o gerenciador de filas locais coloque na fila de transmissão apenas mensagens que o gerenciador de fila parceiro pode processar corretamente.

Esse atributo é principalmente para uso com filas de transmissão, mas o processamento descrito é executado independentemente do uso definido para a fila (consulte [Atributo de uso](#)).

O valor é um dos seguintes:

### **MQDL\_SUPPORTED**

As mensagens da lista de distribuição podem ser armazenadas na fila e transmitidas para o gerenciador de filas de parceria nesse formato Isso reduz a quantidade de processamento necessária para enviar a mensagem para vários destinos

### **MQDL\_NOT\_SUPPORTED**

As mensagens da lista de distribuição não podem ser armazenadas na fila, porque o gerenciador de filas parceiro não suporta listas de distribuição. Se um aplicativo colocar uma mensagem da lista de distribuição e essa mensagem tiver que ser colocada nessa fila, o gerenciador de filas dividirá a mensagem da lista de distribuição e colocará as mensagens individuais na fila. Isso aumenta a

quantia de processamento necessária para enviar a mensagem para diversos destinos, mas assegura que as mensagens sejam processada corretamente pelo gerenciador de filas de parceria

Para determinar o valor desse atributo, use o seletor MQIA\_DIST\_LISTS com a chamada MQINQ.. Para alterar o valor desse atributo, use a chamada MQSET..

Esse atributo não é suportado no z/OS

### **HardenGetRestauração (MQLONG)**

Para cada mensagem, uma contagem é mantida do número de vezes que a mensagem é recuperada por uma chamada MQGET em uma unidade de trabalho e essa unidade de trabalho subsequentemente restaurada.

<i>Tabela 587. Tipos de fila aos quais este atributo se aplica</i>				
Local	Modelo	Alias	Remoto	Cluster
X	X			

Essa contagem está disponível no campo *BackoutCount* no descritor de mensagens após a chamada MQGET ter sido concluída

A contagem de restaurações de mensagens sobrevive a reinicializações do gerenciador de fila No entanto, para assegurar que a contagem seja precisa, as informações devem ser *reforçadas* (registradas no disco ou em outro dispositivo de armazenamento permanente) cada vez que uma chamada MQGET recuperar uma mensagem em uma unidade de trabalho para essa fila. Se isso não for feito, o gerenciador de filas falhará e a chamada MQGET será refeita, a contagem poderá ou não ser incrementada..

O reforço de informações para cada chamada MQGET dentro de uma unidade de trabalho, no entanto, impõe custo de processamento adicional, portanto, configure o atributo **HardenGetBackout** para MQQA\_BACKOUT\_HARDENIDA somente se for essencial que a contagem seja precisa

No Multiplataformas, a contagem de restaurações de mensagens é sempre reforçada, independentemente da configuração desse atributo

Os valores a seguir são possíveis:

#### **MQQA\_BACKOUT\_HARDENED**

O reforço é usado para assegurar que a contagem de restauração para mensagens nesta fila seja precisa.

#### **MQQA\_BACKOUT\_NOT\_HARDENIDOS**

O reforço não é usado para assegurar que a contagem de restauração para mensagens nesta fila seja precisa. Portanto, a contagem pode ser menor do que deveria ser.

Para determinar o valor desse atributo, use o seletor MQIA\_HARDEN\_GET\_BACKOUT com a chamada MQINQ..

### **IndexType (MQLONG)**

Isso especifica o tipo de índice que o gerenciador de filas mantém para mensagens na fila..

<i>Tabela 588. Tipos de fila aos quais este atributo se aplica</i>				
Local	Modelo	Alias	Remoto	Cluster
X	X			

O tipo de índice necessário depende de como o aplicativo recupera mensagens e se a fila é uma fila compartilhada ou uma fila não compartilhada (consulte [atributo QSGDisp](#) ). Os seguintes valores são possíveis para *IndexType*:

#### **MQIT\_NONE**

Nenhum índice é mantido pelo gerenciador de fila para esta fila Use esse valor para filas que são geralmente processadas sequencialmente, ou seja, sem usar nenhum critério de seleção na chamada MQGET

## MQIT\_MSG\_ID

O gerenciador de filas mantém um índice que utiliza os identificadores de mensagens das mensagens na fila. Use essas filas de valores em que o aplicativo geralmente recupera mensagens usando o identificador de mensagens como o critério de seleção na chamada MQGET

## MQIT\_CORREL\_ID

O gerenciador de filas mantém um índice que usa os identificadores de correlação das mensagens na fila.. Use este valor para filas em que o aplicativo geralmente recupera mensagens usando o identificador de correlações como o critério de seleção na chamada MQGET..

## MQIT\_MSG\_TOKEN

**Importante:** Esse tipo de índice deve ser usado apenas para filas usadas com o produto IBM MQ Fluxo de Trabalho para z/OS

O gerenciador de filas mantém um índice que usa os tokens de mensagens das mensagens na fila para uso com as funções do Workload Manager (WLM) do z/OS.

Você *deve* especificar essa opção para filas gerenciadas pelo WLM; não a especifique para nenhum outro tipo de fila.. Além disso, não use esse valor para uma fila em que um aplicativo não está usando as funções do gerenciador de cargas de trabalho do z/OS , mas está recuperando mensagens usando o token de mensagem como um critério de seleção na chamada MQGET

## MQIT\_GROUP\_ID

O gerenciador de filas mantém um índice que usa os identificadores de grupo das mensagens na fila. Esse valor deve ser usado para filas em que o aplicativo recupera mensagens usando a opção MQGMO\_LOGICAL\_ORDER na chamada MQGET.

Uma fila com esse tipo de índice não pode ser uma fila de transmissões. Uma fila compartilhada com esse tipo de índice deve ser definida para ser mapeada para um objeto CFSTRUCT em CFLEVEL (3) ou superior

### Nota:

1. A ordem física de mensagens em uma fila com o tipo de índice MQIT\_GROUP\_ID não está definido, pois a fila é otimizada para recuperação eficiente de mensagens usando a opção MQGMO\_LOGICAL\_ORDER na chamada MQGET. Isso significa que a ordem física das mensagens geralmente não é a ordem na qual as mensagens chegaram à fila
2. Se uma fila MQIT\_GROUP\_ID tiver um *MsgDeliverySequence* de MQMDS\_PRIORITY, o gerenciador de filas usará as prioridades de mensagens 0 e 1 para otimizar a recuperação de mensagens em ordem lógica.. Como resultado, a primeira mensagem em um grupo não deve ter uma prioridade zero ou um; se ela tiver, a mensagem será processada como se tivesse uma prioridade dois. O campo *Priority* na estrutura do MQMD não é alterado

Para obter mais informações sobre grupos de mensagens, consulte a descrição das opções de grupo e de segmento no campo MQGMO-Options

O tipo de índice que deve ser usado em vários casos é mostrado em [Tabela 589 na página 881](#) e [Tabela 590 na página 882](#)..

<i>Tabela 589. Valores sugeridos ou necessários do tipo de índice de fila quando MQGMO_LOGICAL_ORDER não especificado</i>		
<b>Crítérios de seleção na chamada MQGET</b>	<b>Tipo de índice para fila não compartilhada</b>	<b>Tipo de índice para fila compartilhada</b>
Nenhum	Qualquer	Qualquer
<b>Seleção usando um identificador:</b>		
ID da Mensagem	MQIT_MSG_ID sugerido	MQIT_NONE ou MQIT_MSG_ID necessário; MQIT_MSG_ID sugerido
Identificador de correlação	MQIT_CORREL_ID sugerido	MQIT_CORREL_ID necessário

Tabela 589. Valores sugeridos ou necessários do tipo de índice de fila quando MQGMO\_LOGICAL\_ORDER não especificado (continuação)

<b>Crítérios de seleção na chamada MQGET</b>	<b>Tipo de índice para fila não compartilhada</b>	<b>Tipo de índice para fila compartilhada</b>
Identificador de grupo	MQIT_GROUP_ID sugerido	MQIT_GROUP_ID necessário
<b>Seleção usando dois identificadores:</b>		
Identificador de mensagem mais identificador de correlação	MQIT_MSG_ID ou MQIT_CORREL_ID sugerido	MQIT_NONE ou MQIT_MSG_ID ou MQIT_CORREL_ID necessário  (Para eficiência, sugere-se que o tipo de índice seja escolhido para corresponder ao campo MQMD que terá as chaves mais distintas)
Identificador de mensagem mais identificador de grupo	MQIT_MSG_ID ou MQIT_GROUP_ID sugerido	Não Suportado
Identificador de correlação mais identificador de grupo	MQIT_CORREL_ID ou MQIT_GROUP_ID sugerido	Não Suportado
<b>Seleção usando três identificadores:</b>		
Identificador de mensagem mais identificador de correlação mais identificador de grupo	MQIT_MSG_ID ou MQIT_CORREL_ID ou MQIT_GROUP_ID sugerido	Não Suportado
<b>Seleção usando critérios relacionados ao grupo:</b>		
Identificador de grupo mais número de sequência da mensagem	MQIT_GROUP_ID necessário	MQIT_GROUP_ID necessário
Número de sequência da mensagem (deve ser 1).	MQIT_GROUP_ID necessário	MQIT_GROUP_ID necessário
<b>Seleção usando token de mensagem:</b>		
Token de mensagem para uso do aplicativo..	Não usar MQIT_MSG_TOKEN	
Token de mensagem para uso do WLM	MQIT_MSG_TOKEN necessário	Não Suportado


Tabela 590. Valores sugeridos ou necessários do tipo de índice de fila quando MQGMO\_LOGICAL\_ORDER especificado

<b>Crítérios de seleção na chamada MQGET</b>	<b>Tipo de índice para fila não compartilhada</b>	<b>Tipo de índice para fila compartilhada</b>
Nenhum	MQIT_GROUP_ID necessário	MQIT_GROUP_ID necessário
<b>Seleção usando um identificador:</b>		
ID da Mensagem	MQIT_GROUP_ID necessário	Não Suportado
Identificador de correlação	MQIT_GROUP_ID necessário	Não Suportado
Identificador de grupo	MQIT_GROUP_ID necessário	MQIT_GROUP_ID necessário

Tabela 590. Valores sugeridos ou necessários do tipo de índice de fila quando MQGMO\_LOGICAL\_ORDER especificado (continuação)

<b>Crítérios de seleção na chamada MQGET</b>	<b>Tipo de índice para fila não compartilhada</b>	<b>Tipo de índice para fila compartilhada</b>
<b>Seleção usando dois identificadores:</b>		
Identificador de mensagem mais identificador de correlação	MQIT_GROUP_ID necessário	Não Suportado
Identificador de mensagem mais identificador de grupo	MQIT_GROUP_ID necessário	Não Suportado
Identificador de correlação mais identificador de grupo	MQIT_GROUP_ID necessário	Não Suportado
<b>Seleção usando três identificadores:</b>		
Identificador de mensagem mais identificador de correlação mais identificador de grupo	MQIT_GROUP_ID necessário	Não Suportado

Para determinar o valor desse atributo, use o seletor MQIA\_INDEX\_TYPE com a chamada MQINQ..

 Esse atributo é suportado apenas no z/OS

### **InhibitGet (MQLONG)**

Isso controla se operações get para esta fila são permitidas.

Tabela 591. Tipos de fila aos quais este atributo se aplica

<b>Local</b>	<b>Modelo</b>	<b>Alias</b>	<b>Remoto</b>	<b>Cluster</b>
X	X	X		

Se a fila for uma fila de alias, as operações get deverão ser permitidas para o alias e a fila de base no momento da operação get, para que a chamada MQGET seja bem-sucedida. O valor é um dos seguintes:

#### **MQQA\_GET\_INHIBITED**

As operações de obtenção são inibidas

Chamadas MQGET falham com código de razão MQRC\_GET\_INHIBITED. Isso inclui chamadas MQGET que especificam MQGMO\_BROWSE\_FIRST ou MQGMO\_BROWSE\_NEXT..

**Nota:** Se uma chamada MQGET operando em uma unidade de trabalho for concluída com êxito, alterar o valor do atributo **InhibitGet** subsequentemente para MQQA\_GET\_INIBITED não evitará que a unidade de trabalho seja confirmada.

#### **MQQA\_GET\_ALLOWED**

As operações de obtenção são permitidas

Para determinar o valor desse atributo, use o seletor MQIA\_INIBIT\_GET com a chamada MQINQ. Para alterar o valor desse atributo, use a chamada MQSET..

### **InhibitPut (MQLONG)**

Isso controla se operações put para esta fila são permitidas.

Tabela 592. Tipos de fila aos quais este atributo se aplica

<b>Local</b>	<b>Modelo</b>	<b>Alias</b>	<b>Remoto</b>	<b>Cluster</b>
X	X	X	X	X

Se houver mais de uma definição no caminho de resolução de nome da fila, as operações put deverão ser permitidas para *cada* definição no caminho (incluindo quaisquer definições de alias do gerenciador de filas) no momento da operação put, para que a chamada MQPUT ou MQPUT1 seja bem-sucedida. O valor é um dos seguintes:

#### **MQQA\_PUT\_INHIBITED**

Operações put são inibidas.

As chamadas MQPUT e MQPUT1 falham com o código de razão MQRC\_PUT\_INIBIDO.

**Nota:** Se uma chamada MQPUT operando em uma unidade de trabalho for concluída com sucesso, mudar o valor do atributo **InhibitPut** subsequentemente para MQQA\_PUT\_INHIBITED não evitará que a unidade de trabalho seja confirmada.

#### **MQQA\_PUT\_ALLOWED**

Operações put são permitidas.

Para determinar o valor desse atributo, use o seletor MQIA\_INIBIT\_PUT com a chamada MQINQ. Para alterar o valor desse atributo, use a chamada MQSET..

#### **InitiationQName (MQCHAR48)**

Esse é o nome de uma fila definida no gerenciador de fila local; a fila deve ser do tipo MQQT\_LOCAL

<i>Tabela 593. Tipos de fila aos quais este atributo se aplica</i>				
Local	Modelo	Alias	Remoto	Cluster
X				

O gerenciador de filas envia uma mensagem do acionador para a fila de inicialização quando a inicialização do aplicativo é necessária como resultado de uma mensagem que chega na fila à qual esse atributo pertence. A fila de inicialização deve ser monitorada por um aplicativo do monitor acionador que inicia o aplicativo apropriado após o recebimento da mensagem do acionador.

Para determinar o valor desse atributo, use o seletor MQCA\_INITIATION\_Q\_NAME com a chamada MQINQ. O comprimento desse atributo é fornecido por MQ\_Q\_NAME\_LENGTH.

#### **Comprimento de MaxMsg(MQLONG)**

Este é um limite superior para o comprimento da mensagem *física* mais longa que pode ser colocada na fila.

<i>Tabela 594. Tipos de fila aos quais este atributo se aplica</i>				
Local	Modelo	Alias	Remoto	Cluster
X	X			

No entanto, como o atributo da fila **MaxMsgLength** pode ser configurado independentemente do atributo do gerenciador de filas **MaxMsgLength**, o limite superior real para o comprimento da mensagem física mais longa que pode ser colocada na fila é o menor desses dois valores.

Se o gerenciador de filas suportar a segmentação, será possível para um aplicativo colocar uma mensagem *lógica* que seja maior que o menor dos dois atributos **MaxMsgLength**, mas somente se o aplicativo especificar o sinalizador MQMF\_SEGMENTATION\_ALLOWED no MQMD. Se esse sinalizador for especificado, o limite superior para o comprimento de uma mensagem lógica é 999 999 999 bytes, mas geralmente restrições de recursos impostas pelo sistema operacional, ou pelo ambiente no qual o aplicativo está em execução, resultam em um limite inferior.

Uma tentativa de colocar na fila uma mensagem muito longa falha com um dos seguintes códigos de razão:

- MQRC\_MSG\_TOO\_BIG\_FOR\_Q se a mensagem for muito grande para a fila
- MQRC\_MSG\_TOO\_BIG\_FOR\_Q\_MGR se a mensagem for muito grande para o gerenciador de filas, mas não muito grande para a fila



O limite inferior para o atributo **MaxMsgLength** é zero; o limite superior é 100 MB (104 857 600 bytes).

Para obter mais informações, consulte [MQPUT-parâmetro BufferLength](#)

Para determinar o valor desse atributo, use o seletor MQIA\_MAX\_MSG\_LENGTH com a chamada MQINQ..

### **MaxQDepth (MQLONG)**

Este é o limite superior definido para o número de mensagens físicas que podem existir na fila a qualquer momento

Tabela 595. Tipos de fila aos quais este atributo se aplica				
Local	Modelo	Alias	Remoto	Cluster
X	X			

Uma tentativa de colocar uma mensagem na fila que já contém mensagens **MaxQDepth** falha com o código de razão MQRC\_Q\_FULL.

O processamento da unidade de trabalho e a segmentação de mensagens podem fazer o número real de mensagens físicas na fila exceder **MaxQDepth**. No entanto, isso não afeta a recuperabilidade da mensagem porque todas as mensagens na fila podem ser recuperadas usando a chamada MQGET.

O valor deste atributo é zero ou maior. O limite superior é determinado pelo ambiente:

- Nas plataformas a seguir, o valor não pode exceder 999 999 999:

-  AIX
-  Linux
-  Windows
-  z/OS

-  No IBM i, o valor não pode exceder 640 000.

**Nota:** O espaço de armazenamento disponível para a fila pode ser esgotado mesmo se houver menos de **MaxQDepth** mensagens na fila.

Para determinar o valor desse atributo, use o seletor MQIA\_MAX\_Q\_DEPTH com a chamada MQINQ..

### **MsgDeliverySequência (MQLONG)**

Tabela 596. Tipos de fila aos quais este atributo se aplica				
Local	Modelo	Alias	Remoto	Cluster
X	X			

Isso determina a ordem na qual a chamada MQGET retorna mensagens para o aplicativo:

#### **MQMDS\_FIFO**

Mensagens são retornadas na ordem FIFO (primeiro a entrar, primeiro a sair).

Uma chamada MQGET retorna a *primeira* mensagem que satisfaz os critérios de seleção especificados na chamada, independentemente da prioridade da mensagem.

#### **MQMDS\_PRIORITY**

As mensagens são retornadas em ordem de prioridade

Uma chamada MQGET retorna a mensagem *de prioridade mais alta* que atende aos critérios de seleção especificados na chamada.. Dentro de cada nível de prioridade, as mensagens são retornadas na ordem FIFO (primeiro a entrar, primeiro a sair)

- No z/OS, se a fila tiver um *IndexType* de MQIT\_GROUP\_ID, o atributo **MsgDeliverySequence** especifica a ordem na qual os grupos de mensagens são retornados para o aplicativo. A sequência

específica na qual os grupos são retornados é determinada pela posição ou prioridade da primeira mensagem em cada grupo. A ordem física de mensagens na fila não é definida, pois a fila é otimizada para recuperação eficiente de mensagens usando a opção MQGMO\_LOGICAL\_ORDER na chamada MQGET..

- No z/OS, se *IndexType* for MQIT\_GROUP\_ID e *MsgDeliverySequence* for MQMDS\_PRIORITY, o gerenciador de filas usará as prioridades de mensagens zero e um para otimizar a recuperação de mensagens na ordem lógica... Como resultado, a primeira mensagem em um grupo não deve ter uma prioridade zero ou um; se ela tiver, a mensagem será processada como se tivesse uma prioridade dois. O campo *Priority* na estrutura do MQMD não é alterado

Se os atributos relevantes forem alterados enquanto houver mensagens na fila, a sequência de entrega será a seguinte:

- A ordem na qual as mensagens são retornadas pela chamada MQGET é determinada pelos valores dos atributos **MsgDeliverySequence** e **DefPriority** em vigor para a fila no momento em que a mensagem chega à fila:
  - Se *MsgDeliverySequence* for MQMDS\_FIFO quando a mensagem chegar, ela será colocada na fila como se sua prioridade fosse *DefPriority*. Isso não afeta o valor do campo *Priority* no descritor de mensagens da mensagem; esse campo retém o valor que ele tinha quando a mensagem foi colocada pela primeira vez
  - Se *MsgDeliverySequence* for MQMDS\_PRIORITY quando a mensagem chegar, a mensagem será colocada na fila no local apropriado para a prioridade fornecida pelo campo *Priority* no descritor de mensagens

Se o valor do atributo **MsgDeliverySequence** for alterado enquanto houver mensagens na fila, a ordem das mensagens na fila não será alterada.

Se o valor do atributo **DefPriority** for alterado enquanto houver mensagens na fila, as mensagens não serão necessariamente entregues na ordem FIFO, mesmo que o atributo **MsgDeliverySequence** seja configurado como MQMDS\_FIFO; aqueles que foram colocados na fila na prioridade mais alta serão entregues primeiro..

Para determinar o valor desse atributo, use o seletor MQIA\_MSG\_DELIVERY\_SEQUENCE com a chamada MQINQ

### **NonPersistentMessageClass (MQLONG)**

O objetivo de confiabilidade para mensagens não persistentes

*Tabela 597. Tipos de fila aos quais este atributo se aplica*

Local	Modelo	Alias	Remoto	Cluster
X	X			

Isso especifica as circunstâncias nas quais as mensagens não persistentes colocadas nesta fila são descartadas:

#### **MQNPM\_CLASS\_NORMAL**

As mensagens não persistentes são limitadas ao tempo de vida da sessão do gerenciador de filas; as mensagens são descartadas no evento de uma reinicialização do gerenciador de filas. Isso é válido apenas para filas não compartilhadas e é o valor padrão.

#### **MQNPM\_CLASS\_HIGH**

O gerenciador de filas tenta reter mensagens não persistentes durante o tempo de vida da fila. As mensagens não persistentes ainda podem ser perdidas no caso de uma falha. Esse valor é aplicado para filas compartilhadas.

Para determinar o valor desse atributo, use o seletor MQIA\_NPM\_CLASS com a chamada MQINQ..

### **Contagem de OpenInput(MQLONG)**

Este é o número de identificadores que são atualmente válidos para remover mensagens da fila por meio da chamada MQGET.

<i>Tabela 598. Tipos de fila aos quais este atributo se aplica</i>				
<b>Local</b>	<b>Modelo</b>	<b>Alias</b>	<b>Remoto</b>	<b>Cluster</b>
X				

É o número total desses identificadores conhecidos no gerenciador de filas *local*. Se a fila for uma fila compartilhada, a contagem não incluirá aberturas para entrada que foram executadas para a fila em outros gerenciadores de fila no grupo de filas compartilhadas ao qual o gerenciador de fila local pertence..

A contagem inclui identificadores nos quais uma fila de alias resolvida para essa fila foi aberta para entrada. A contagem não inclui identificadores nos quais a fila foi aberta para ações que não incluíram entrada (por exemplo, uma fila aberta apenas para navegação)..

O valor desse atributo flutua conforme o gerenciador de filas opera.

Para determinar o valor desse atributo, use o seletor MQIA\_OPEN\_INPUT\_COUNT com a chamada MQINQ..

### **Contagem de OpenOutput(MQLONG)**

Este é o número de identificadores que são atualmente válidos para incluir mensagens na fila por meio de chamada MQPUT

<i>Tabela 599. Tipos de fila aos quais este atributo se aplica</i>				
<b>Local</b>	<b>Modelo</b>	<b>Alias</b>	<b>Remoto</b>	<b>Cluster</b>
X				

É o número total desses identificadores conhecidos do gerenciador de filas *local*; ele não inclui aberturas para saída que foram executadas para essa fila em gerenciadores de filas remotas. Se a fila for uma fila compartilhada, a contagem não incluirá aberturas para a saída que foram executadas para a fila em outros gerenciadores de filas no grupo de filas compartilhadas ao qual o gerenciador de filas locais pertence

A contagem inclui identificadores onde uma fila de alias que resolve para essa fila foi aberta para saída. A contagem não inclui identificadores nos quais a fila foi aberta para ações que não incluíram saída (por exemplo, uma fila aberta somente para consulta)..

O valor desse atributo flutua conforme o gerenciador de filas opera.

Para determinar o valor desse atributo, use o seletor MQIA\_OPEN\_OUTPUT\_COUNT com a chamadas MQINQ.

### **ProcessName (MQCHAR48)**

Este é o nome de um objeto de processo definido no gerenciador de fila local. O objeto de processo identifica um programa que pode atender a fila..

<i>Tabela 600. Tipos de fila aos quais este atributo se aplica</i>				
<b>Local</b>	<b>Modelo</b>	<b>Alias</b>	<b>Remoto</b>	<b>Cluster</b>
X	X			

Para determinar o valor desse atributo, use o seletor MQCA\_PROCESS\_NAME com a chamada MQINQ.. O comprimento desse atributo é fornecido por MQ\_PROCESS\_NAME\_LENGTH

## PropertyControl (MQLONG)

Especifica como as propriedades de mensagens são manipuladas para mensagens que são recuperadas de filas usando a chamada MQGET com a opção MQGMO\_PROPERTIES\_AS\_Q\_DEF.

Tabela 601. Tipos de fila aos quais este atributo se aplica				
Local	Modelo	Alias	Remoto	Cluster
X	X	X		

O valor é um dos seguintes:

### MQPROP\_ALL

Todas as propriedades da mensagem são incluídas na mensagem quando ela é entregue ao aplicativo. As propriedades, exceto aquelas no descritor de mensagens (ou extensão), são colocadas em um ou mais cabeçalhos MQRFH2 nos dados da mensagem. Se um identificador de mensagem for fornecido, o comportamento será retornar as propriedades no identificador de mensagens

### MQPROP\_COMPATIBILITY

Se a mensagem contiver uma propriedade com um prefixo de mcd., jms., usr. ou mqext., todas as propriedades de mensagem são entregues ao aplicativo em um cabeçalho MQRFH2. Caso contrário, todas as propriedades da mensagem, exceto aquelas contidas no descritor de mensagens (ou extensão), serão descartadas e não serão mais acessíveis ao aplicativo. Este é o valor padrão; ele permite que os aplicativos que esperam que as propriedades relacionadas ao JMS estejam em um cabeçalho MQRFH2 nos dados da mensagem continuem funcionando sem modificação. Se um identificador de mensagem for fornecido, então o comportamento será retornar as propriedades no identificador de mensagens ...

### MQPROP\_FORCE\_MQRFH2

Propriedades são sempre retornadas nos dados da mensagem em um cabeçalho MQRFH2, independentemente de o aplicativo especificar um identificador de mensagens. Uma manipulação de mensagem válida fornecida no campo MsgHandle da estrutura MQGMO na chamada MQGET é ignorada. As propriedades da mensagem não são acessíveis por meio do identificador de mensagem.

### MQPROP\_NONE

Todas as propriedades da mensagem, exceto aquelas no descritor de mensagens (ou extensão), são removidas da mensagem antes que a mensagem seja entregue ao aplicativo. Se um identificador de mensagem for fornecido, o comportamento será retornar as propriedades no identificador de mensagens

Esse parâmetro é aplicável às filas Local, Alias e Modelo. Para determinar seu valor, use o seletor MQIA\_PROPERTY\_CONTROL com a chamada MQINQ.

## Evento QDepthHigh(MQLONG)

Isso controla se os eventos de Alta Profundidade da Fila são gerados

Tabela 602. Tipos de fila aos quais este atributo se aplica				
Local	Modelo	Alias	Remoto	Cluster
X	X			

Um evento de Profundidade de Fila Alta indica que um aplicativo colocou uma mensagem em uma fila e isso fez com que o número de mensagens na fila se torne maior ou igual ao limite alto de profundidade da fila (consulte o atributo **QDepthHighLimit**).

**Nota:** O valor desse atributo pode mudar dinamicamente.

O valor é um dos seguintes:

### MQEVR\_DISABLED

Relatório de eventos desativado.

### MQEVR\_ENABLED

Relatório de eventos ativado

Para obter mais informações sobre os eventos, consulte [Monitoramento de eventos](#)

Para determinar o valor desse atributo, use o seletor MQIA\_Q\_DEPTH\_HIGH\_EVENT com a chamada MQINQ

Esse atributo é suportado no z/OS, mas a chamada MQINQ não pode ser usada para determinar seu valor..

### **QDepthHighLimite (MQLONG)**

Este é o limite com relação ao qual a profundidade da fila é comparada para gerar um evento de Alta Profundidade da fila...

<i>Tabela 603. Tipos de fila aos quais este atributo se aplica</i>				
<b>Local</b>	<b>Modelo</b>	<b>Alias</b>	<b>Remoto</b>	<b>Cluster</b>
X	X			

Este evento indica que um aplicativo colocou uma mensagem em uma fila e que isso fez com que o número de mensagens na fila se tornasse maior ou igual ao limite alto de profundidade da fila.. Consulte [QDepthHighAtributo](#) do evento..

O valor é expresso como uma porcentagem da profundidade máxima da fila (atributo **MaxQDepth**) e é maior ou igual a 0 e menor ou igual a 100. O valor padrão é 80.

Para determinar o valor desse atributo, use o seletor MQIA\_Q\_DEPTH\_HIGH\_LIMIT com a chamada MQINQ..

Esse atributo é suportado no z/OS, mas a chamada MQINQ não pode ser usada para determinar seu valor..

### **Evento QDepthLow(MQLONG)**

Isso controla se os eventos de Baixa Profundidade da Fila são gerados ou não.

<i>Tabela 604. Tipos de fila aos quais este atributo se aplica</i>				
<b>Local</b>	<b>Modelo</b>	<b>Alias</b>	<b>Remoto</b>	<b>Cluster</b>
X	X			

Um evento de Baixa Profundidade da Fila indica que um aplicativo recuperou uma mensagem de uma fila e que isso fez com que o número de mensagens na fila se tornasse menor ou igual ao limite de baixa profundidade da fila (veja [QDepthLowAtributo Limite](#)).

**Nota:** O valor desse atributo pode mudar dinamicamente.

O valor é um dos seguintes:

#### **MQEVR\_DISABLED**

Relatório de eventos desativado.

#### **MQEVR\_ENABLED**

Relatório de eventos ativado

Para obter mais informações sobre os eventos, consulte [Monitoramento de eventos](#)

Para determinar o valor desse atributo, use o seletor MQIA\_Q\_DEPTH\_LOW\_EVENT com a chamado MQINQ.

Esse atributo é suportado no z/OS, mas a chamada MQINQ não pode ser usada para determinar seu valor..

### **Limite QDepthLow(MQLONG)**

Este é o limite com relação ao qual a profundidade da fila é comparada para gerar um evento de Baixa Profundidade da fila...

<i>Tabela 605. Tipos de fila aos quais este atributo se aplica</i>				
<b>Local</b>	<b>Modelo</b>	<b>Alias</b>	<b>Remoto</b>	<b>Cluster</b>
X	X			

Esse evento indica que um aplicativo recuperou uma mensagem de uma fila e que isso fez com que o número de mensagens na fila se tornasse menor ou igual ao limite baixo de profundidade da fila Consulte [QDepthLow](#) atributo do evento.

O valor é expresso como uma porcentagem da profundidade máxima da fila (atributo **MaxQDepth**) e é maior ou igual a 0 e menor ou igual a 100. O valor padrão é 20.

Para determinar o valor desse atributo, use o seletor MQIA\_Q\_DEPTH\_LOW\_LIMIT com a chamada MQINQ..

Esse atributo é suportado no z/OS, mas a chamada MQINQ não pode ser usada para determinar seu valor..

### **Evento QDepthMax(MQLONG)**

Isso controla se os eventos de Fila Cheia são gerados Um evento Fila Cheia indica que um put para uma fila foi rejeitado porque a fila está cheia, ou seja, a profundidade da fila já atingiu seu valor máximo.

<i>Tabela 606. Tipos de fila aos quais este atributo se aplica</i>				
<b>Local</b>	<b>Modelo</b>	<b>Alias</b>	<b>Remoto</b>	<b>Cluster</b>
X	X			

**Nota:** O valor desse atributo pode mudar dinamicamente.

O valor é um dos seguintes:

#### **MQEVR\_DISABLED**

Relatório de eventos desativado.

#### **MQEVR\_ENABLED**

Relatório de eventos ativado

Para obter mais informações sobre os eventos, consulte [Monitoramento de eventos](#)

Para determinar o valor desse atributo, use o seletor MQIA\_Q\_DEPTH\_MAX\_EVENT com a chamada MQINQ..

Esse atributo é suportado no z/OS, mas a chamada MQINQ não pode ser usada para determinar seu valor..

### **QDesc (MQCHAR64)**

Use este campo para comentário descritivo

<i>Tabela 607. Tipos de fila aos quais este atributo se aplica</i>				
<b>Local</b>	<b>Modelo</b>	<b>Alias</b>	<b>Remoto</b>	<b>Cluster</b>
X	X	X	X	X

O conteúdo do campo não é significativo para o gerenciador de filas, mas o gerenciador de filas pode requerer que o campo contenha apenas caracteres que possam ser exibidos... Ele não pode conter nenhum caractere nulo; se necessário, será preenchido à direita com espaços em branco. Em uma instalação DBCS, o campo pode conter caracteres DBCS (sujeito a um comprimento máximo de campo de 64 bytes).

**Nota:** Se esse campo contiver caracteres que não estão no conjunto de caracteres do gerenciador de filas (conforme definido pelo atributo do gerenciador de filas **CodedCharSetId**), esses caracteres poderão ser convertidos incorretamente se esse campo for enviado para outro gerenciador de filas..

Para determinar o valor desse atributo, use o seletor MQCA\_Q\_DESC com a chamadas MQINQ. O comprimento desse atributo é fornecido por MQ\_Q\_DESC\_LENGTH

### **QName (MQCHAR48)**

Este é o nome de uma fila definida no Gerenciador de Filas Local

<i>Tabela 608. Tipos de fila aos quais este atributo se aplica</i>				
<b>Local</b>	<b>Modelo</b>	<b>Alias</b>	<b>Remoto</b>	<b>Cluster</b>
X		X	X	X

Todas as filas definidas em um gerenciador de filas compartilham o mesmo namespace da fila. Portanto, uma fila MQQT\_LOCAL e uma fila MQQT\_ALIAS não podem ter o mesmo nome.

Para determinar o valor desse atributo, use o seletor MQCA\_Q\_NAME com a chamada MQINQ.. O comprimento desse atributo é fornecido por MQ\_Q\_NAME\_LENGTH

### **QServiceInterval (MQLONG)**

Este é o intervalo de serviço usado para comparação para gerar eventos de Intervalo de Serviço Alto e de Intervalo de Serviço OK

<i>Tabela 609. Tipos de fila aos quais este atributo se aplica</i>				
<b>Local</b>	<b>Modelo</b>	<b>Alias</b>	<b>Remoto</b>	<b>Cluster</b>
X	X			

Consulte [QServiceIntervalAtributo](#) do evento.

O valor está em unidades de milissegundos e é maior ou igual a zero e menor ou igual a 999 999 999.

Para determinar o valor desse atributo, use o seletor MQIA\_Q\_SERVICE\_INTERVAL com a chamada MQINQ

Esse atributo é suportado no z/OS, mas a chamada MQINQ não pode ser usada para determinar seu valor..

### **QServiceIntervalEvent (MQLONG)**

Isso controla se os eventos Service Interval High ou Service Interval OK são gerados.

<i>Tabela 610. Tipos de fila aos quais este atributo se aplica</i>				
<b>Local</b>	<b>Modelo</b>	<b>Alias</b>	<b>Remoto</b>	<b>Cluster</b>
X	X			

- Um evento Alto Intervalo de Serviço é gerado quando uma verificação indica que nenhuma mensagem foi recuperada da fila pelo menos pelo tempo indicado pelo atributo **QServiceInterval** .
- Um evento Service Interval OK é gerado quando uma verificação indica que as mensagens foram recuperadas da fila dentro do tempo indicado pelo atributo **QServiceInterval** .

**Nota:** O valor desse atributo pode mudar dinamicamente.

O valor é um dos seguintes:

#### **MQQSIE\_HIGH**

Eventos de Alto Intervalo de Serviço da Fila ativados.

- Eventos de Alto Intervalo de Serviço da Fila são **ativados** e
- Os eventos OK do Intervalo de Serviço da Fila são **desativados**.

#### **MQQSIE\_OK**

Eventos de intervalo de serviço de fila OK ativados.

- Os eventos Alto Intervalo de Serviço da Fila são **desativados** e
- Eventos OK do Intervalo de Serviço da Fila são **ativados**.

### **MQQSIE\_NONE**

Nenhum evento de intervalo de serviço de filas ativado

- Os eventos Alto Intervalo de Serviço da Fila são **desativados** e
- Os eventos de Intervalo de Serviço de Fila OK também são **desativados**.

Para filas compartilhadas, o valor desse atributo é ignorado; o valor MQQSIE\_NONE é assumido.

Para obter mais informações sobre os eventos, consulte [Monitoramento de eventos](#)

Para determinar o valor desse atributo, use o seletor MQIA\_Q\_SERVICE\_INTERVAL\_EVENT com a chamada MQINQ..

No z/OS, não é possível usar a chamada MQINQ para determinar o valor desse atributo...

### **QSGDisp (MQLONG)**

Isso especifica a disposição da fila

<i>Tabela 611. Tipos de fila aos quais este atributo se aplica</i>				
<b>Local</b>	<b>Modelo</b>	<b>Alias</b>	<b>Remoto</b>	<b>Cluster</b>
X		X	X	

O valor é um dos seguintes:

#### **MQQSGD\_Q\_MGR**

O objeto tem a disposição do gerenciador de filas Isso significa que a definição de objeto é conhecida apenas para o gerenciador de filas locais; a definição não é conhecida para outros gerenciadores de filas no grupo de filas compartilhadas

Cada gerenciador de filas no grupo de filas compartilhadas pode ter um objeto com o mesmo nome e tipo que o objeto atual, mas eles são objetos separados e não há correlação entre eles Seus atributos não são restritos a serem iguais entre si.

#### **MQQSGD\_COPY**

O objeto é uma cópia local de uma definição de objeto principal que existe no repositório compartilhado. Cada gerenciador de filas no grupo de filas compartilhadas pode ter a sua própria cópia do objeto. Inicialmente, todas as cópias têm os mesmos atributos, mas usando comandos MQSC, é possível alterar cada cópia para que seus atributos sejam diferentes daqueles das outras cópias.. Os atributos das cópias são ressincronizados quando a definição principal no repositório compartilhado é alterado..

#### **MQQSGD\_SHARED**

O objeto possui uma disposição compartilhada Isso significa que existe no repositório compartilhado uma única instância do objeto conhecido por todos os gerenciadores de filas no grupo de filas compartilhadas. Quando um gerenciador de filas no grupo acessa o objeto, ele acessa a única instância compartilhada do objeto

Para determinar o valor desse atributo, use o seletor MQIA\_QSG\_DISP com a chamada MQINQ.

 Esse atributo é suportado apenas no z/OS

### **QueueAccounting (MQLONG)**

<i>Tabela 612. Tipos de fila aos quais este atributo se aplica</i>				
<b>Local</b>	<b>Modelo</b>	<b>Alias</b>	<b>Remoto</b>	<b>Cluster</b>
X	X	X	X	



Isso controla a coleta de dados de contabilidade para a fila. Para que os dados de contabilidade sejam coletados para essa fila, os dados de contabilidade para essa conexão também devem ser ativados, usando o atributo ACCTQ do QMGR ou o campo Opções na estrutura MQCNO na chamada MQCONN.

Este atributo possui um dos seguintes valores:

**MQMON\_Q\_MGR**

Os dados de contabilidade para essa fila são coletados com base na configuração do atributo ACCTQ do QMGR. Essa é a configuração padrão.

**MQMON\_OFF**

Não coletar dados de contabilidade para esta fila.

**MQMON\_ON**

Coletar dados de contabilidade para esta fila.

Para determinar o valor desse atributo, use o seletor MQIA\_ACCOUNTING\_Q com a chamada MQINQ..

**QueueMonitoring (MQLONG)**

Controla a coleção de dados de monitoramento on-line para filas.

<i>Tabela 613. Tipos de fila aos quais este atributo se aplica</i>				
Local	Modelo	Alias	Remoto	Cluster
X	X			

O valor é um dos seguintes:

**MQMON\_Q\_MGR**

Colete dados de monitoramento de acordo com a configuração do atributo do gerenciador de filas do **QueueMonitoring** Esse é o valor-padrão.

**MQMON\_OFF**

A coleta de dados de monitoramento on-line está desativada para esta fila

**MQMON\_LOW**

Se o valor do atributo do gerenciador de filas do **QueueMonitoring** não for MQMON\_NONE, a coleta de dados de monitoramento on-line será ativada, com uma taxa baixa de coleta de dados para essa fila...

**MQMON\_MEDIUM**

Se o valor do atributo do gerenciador de filas **QueueMonitoring** não for MQMON\_NONE, a coleta de dados de monitoramento on-line será ativada, com uma taxa moderada de coleta de dados para essa fila.

**MQMON\_HIGH**

Se o valor do atributo do gerenciador de filas **QueueMonitoring** não for MQMON\_NONE, a coleta de dados de monitoramento on-line será ativada, com uma alta taxa de coleta de dados para essa fila.

Para determinar o valor desse atributo, use o seletor MQIA\_MONITORING\_Q com a chamada MQINQ..

**QueueStatistics (MQCHAR12)**

<i>Tabela 614. Tipos de fila aos quais este atributo se aplica</i>				
Local	Modelo	Alias	Remoto	Cluster
X	X	X	X	

Isso controla a coleção de dados de estatísticas para a fila

Este atributo possui um dos seguintes valores:

## **MQMON\_Q\_MGR**

Dados contábeis para essa fila são coletados com base na configuração do atributo QMGR STATQ. Essa é a configuração padrão.

## **MQMON\_OFF**

Desative a coleta de dados de estatísticas para esta fila

## **MQMON\_ON**

Ative a coleta de dados estatísticos para essa fila

## **QType (MQLONG)**

<i>Tabela 615. Tipos de fila aos quais este atributo se aplica</i>				
<b>Local</b>	<b>Modelo</b>	<b>Alias</b>	<b>Remoto</b>	<b>Cluster</b>
X		X	X	X

Este é o tipo de fila; ele possui um dos seguintes valores:

### **MQQT\_ALIAS**

Definição de fila de alias

### **MQQT\_CLUSTER**

Fila de clusters

### **MQQT\_LOCAL**

Fila local.

### **MQQT\_REMOTE**

A definição local de uma fila remota

Para determinar o valor desse atributo, use o seletor MQIA\_Q\_TYPE com a chamada MQINQ.

## **RemoteQMgrNome (MQCHAR48)**

<i>Tabela 616. Tipos de fila aos quais este atributo se aplica</i>				
<b>Local</b>	<b>Modelo</b>	<b>Alias</b>	<b>Remoto</b>	<b>Cluster</b>
			X	

Este é o nome do gerenciador de filas remotas no qual a fila **RemoteQName** está definida. Se a fila **RemoteQName** tiver um valor **QSGDisp** de MQQSGD\_COPY ou MQQSGD\_SHARED, **RemoteQMgrName** poderá ser o nome do grupo de filas compartilhadas que possui **RemoteQName**.

Se um aplicativo abrir a definição local de uma fila remota, **RemoteQMgrName** não deverá estar em branco e não deverá ser o nome do gerenciador de fila local. Se **XmitQName** estiver em branco, a fila local com o mesmo nome que **RemoteQMgrName** será usada como a fila de transmissão. Se não houver nenhuma fila com o nome **RemoteQMgrName**, a fila identificada pelo atributo do gerenciador de filas **DefXmitQName** será usada.

Se essa definição for usada para um alias do gerenciador de filas, **RemoteQMgrName** será o nome do gerenciador de filas que está sendo alias.... Ele pode ser o nome do gerenciador de fila local. Caso contrário, se **XmitQName** estiver em branco quando a abertura ocorrer, deverá haver uma fila local com um nome igual ao **RemoteQMgrName**; essa fila será usada como a fila de transmissão.

Se essa definição for usada para um alias de resposta, esse nome será o nome do gerenciador de filas que deve ser o **ReplyToQMgr**.

**Nota:** Nenhuma validação é executada no valor especificado para este atributo quando a definição de fila é criada ou modificada..

Para determinar o valor desse atributo, use o seletor MQCA\_REMOTE\_Q\_MGR\_NAME com a chamada MQINQ.. O comprimento desse atributo é fornecido por MQ\_Q\_MGR\_NAME\_LENGTH

## RemoteQName (MQCHAR48)

Tabela 617. Tipos de fila aos quais este atributo se aplica				
Local	Modelo	Alias	Remoto	Cluster
			X	

Este é o nome da fila como ele é conhecido no gerenciador de filas remotas *RemoteQMGrName*

Se um aplicativo abrir a definição local de uma fila remota, quando a abertura ocorrer *RemoteQName* não deverá ficar em branco.

Se essa definição for usada para uma definição de alias do gerenciador de fila, quando a abertura ocorrer, *RemoteQName* deverá estar em branco

Se a definição for usada para um alias de resposta, esse nome será o nome da fila que deve ser o *ReplyToQ*

**Nota:** Nenhuma validação é executada no valor especificado para este atributo quando a definição de fila é criada ou modificada..

Para determinar o valor desse atributo, use o seletor MQCA\_REMOTE\_Q\_NAME com a chamada MQINQ O comprimento desse atributo é fornecido por MQ\_Q\_NAME\_LENGTH

## RetentionInterval (MQLONG)

Este é o período de tempo para o qual reter a fila.. Após esse tempo, a fila será elegível para exclusão.

Tabela 618. Tipos de fila aos quais este atributo se aplica				
Local	Modelo	Alias	Remoto	Cluster
X	X			

O tempo é medido em horas contando a partir da data e hora em que a fila foi criada. A data e a hora de criação da fila são registradas nos atributos **CreationDate** e **CreationTime**

Essas informações são fornecidas para permitir que um aplicativo de manutenção ou o operador identifique e exclua filas que não são mais necessárias.

**Nota:** O gerenciador de filas nunca toma nenhuma ação para excluir filas baseadas nesse atributo ou para evitar a exclusão de filas com um intervalo de retenção que não expirou; é responsabilidade do usuário tomar qualquer ação necessária.

Use um intervalo de retenção realista para evitar a acumulação de filas dinâmicas permanentes (consulte o atributo *DefinitionType*). No entanto, esse atributo também pode ser usado com filas predefinidas

Para determinar o valor desse atributo, use o seletor MQIA\_RETENTION\_INTERVAL com a chamada MQINQ

## Escopo (MQLONG)

Isso controla se uma entrada para essa fila também existe em um diretório de célula..

Tabela 619. Tipos de fila aos quais este atributo se aplica				
Local	Modelo	Alias	Remoto	Cluster
X		X	X	

Um diretório de célula é fornecido por um serviço de Nome instalável O valor é um dos seguintes:

### MQSCO\_Q\_MGR

A definição de fila tem o escopo do gerenciador de filas: a definição da fila não se estende além do gerenciador de filas que o possui Para abrir a fila para saída de algum outro gerenciador de filas, o

nome do gerenciador de filas proprietário deve ser especificado ou o outro gerenciador de filas deve ter uma definição local da fila.

### **MQSCO\_CELL**

A definição de fila tem escopo de célula: a definição de fila também é colocada em um diretório de célula disponível para todos os gerenciadores de fila na célula. A fila pode ser aberta para saída de qualquer um dos gerenciadores de filas na célula, especificando o nome da fila; o nome do gerenciador de filas que possui a fila não precisa ser especificado. No entanto, a definição de fila não está disponível para nenhum gerenciador de fila na célula que também tenha uma definição local de uma fila com esse nome, pois a definição local tem precedência..

Um diretório de célula é fornecido por um serviço de Nome instalável

O modelo e as filas dinâmicas não podem ter escopo de célula

Este valor será válido apenas se um serviço de nomes que suporta um diretório de célula tiver sido configurado..

Para determinar o valor desse atributo, use o seletor MQIA\_SCOPE com a chamada MQINQ.

O suporte para esse atributo está sujeito às seguintes restrições:

- No IBM i, o atributo é suportado, mas somente MQSCO\_Q\_MGR é válido
- No z/OS, o atributo não é suportado

### **Compartilhamento (MQLONG)**

Isso indica se a fila pode ser aberta para entrada várias vezes simultaneamente.

<i>Tabela 620. Tipos de fila aos quais este atributo se aplica</i>				
<b>Local</b>	<b>Modelo</b>	<b>Alias</b>	<b>Remoto</b>	<b>Cluster</b>
X	X			

O valor é um dos seguintes:

#### **MQQA\_SHAREABLE**

A fila é compartilhável

Várias aberturas com a opção MQOO\_INPUT\_SHARED são permitidas

#### **MQQA\_NOT\_SHAREABLE**

A fila não é compartilhável

Uma chamada MQOPEN com a opção MQOO\_INPUT\_SHARED é tratada como MQOO\_INPUT\_EXCLUSIVE..

Para determinar o valor deste atributo, use o seletor MQIA\_SHAREABILITY com a chamada MQINQ.

### **StorageClass (MQCHAR8)**

Este é um nome definido pelo usuário que define o armazenamento físico usado para reter a fila. Na prática, uma mensagem é gravada no disco somente se ela precisar ser paginada fora de seu buffer de memória.

<i>Tabela 621. Tipos de fila aos quais este atributo se aplica</i>				
<b>Local</b>	<b>Modelo</b>	<b>Alias</b>	<b>Remoto</b>	<b>Cluster</b>
X	X			

Para determinar o valor desse atributo, use o seletor MQCA\_STORAGE\_CLASS com a chamada MQINQ.. O comprimento desse atributo é fornecido por MQ\_STORAGE\_CLASS\_LENGTH

 Esse atributo é suportado apenas no z/OS

### **TriggerControl (MQLONG)**

Isso controla se mensagens do acionador são gravadas em uma fila de inicialização para iniciar um aplicativo para atender a fila.

Tabela 622. Tipos de fila aos quais este atributo se aplica				
Local	Modelo	Alias	Remoto	Cluster
X	X			

Essa ação é uma das seguintes:

#### **MQTC\_OFF**

Nenhuma mensagem do acionador deve ser gravada para esta fila O valor *TriggerType* é irrelevante neste caso.

#### **MQTC\_ON**

As mensagens do acionador devem ser gravadas para esta fila quando os eventos do acionador apropriados ocorrerem

Para determinar o valor desse atributo, use o seletor MQIA\_TRIGGER\_CONTROL com a chamada MQINQ.. Para alterar o valor desse atributo, use a chamada MQSET..

### **TriggerData (MQCHAR64)**

Estes são dados de formato livre que o gerenciador de filas insere na mensagem do acionador quando uma mensagem que chega a essa fila faz com que uma mensagem do acionador seja gravada na fila de inicialização

Tabela 623. Tipos de fila aos quais este atributo se aplica				
Local	Modelo	Alias	Remoto	Cluster
X	X			

O conteúdo desses dados não é significativo para o gerenciador da fila. Ele é significativo para o aplicativo acionador-monitor que processa a fila de iniciação ou para o aplicativo que o monitor acionador inicia

A sequência de caracteres não deve conter nenhum valor nulo. É preenchido à direita com espaços em branco, se necessário.

Para determinar o valor desse atributo, use o seletor MQCA\_TRIGGER\_DATA com a chamada MQINQ.. Para alterar o valor desse atributo, use a chamada MQSET.. O comprimento deste atributo é fornecido por MQ\_TRIGGER\_DATA\_LENGTH

### **TriggerDepth (MQLONG)**

Tabela 624. Tipos de fila aos quais este atributo se aplica				
Local	Modelo	Alias	Remoto	Cluster
X	X			

Este é o número de mensagens de prioridade *TriggerMsgPriority* ou maior que devem estar na fila antes de uma mensagem do acionador ser gravada Isso se aplica quando o *TriggerType* é configurado como MQTT\_DEPTH O valor de *TriggerDepth* é um ou maior. Caso contrário, esse atributo não será usado

Para determinar o valor desse atributo, use o seletor MQIA\_TRIGGER\_DEPTH com a chamada MQINQ.. Para alterar o valor desse atributo, use a chamada MQSET..

### **TriggerMsgPrioridade (MQLONG)**

Essa é a prioridade da mensagem abaixo da qual as mensagens não contribuem para a geração de mensagens do acionador (ou seja, o gerenciador de filas ignora essas mensagens ao decidir se deve gerar uma mensagem do acionador).

Tabela 625. Tipos de fila aos quais este atributo se aplica				
Local	Modelo	Alias	Remoto	Cluster
X	X			

*TriggerMsgPriority* pode estar no intervalo zero (mais baixo) a *MaxPriority* (mais alto; consulte *MaxPriority*); um valor zero faz com que todas as mensagens contribuam para a geração de mensagens do acionador.

Para determinar o valor desse atributo, use o seletor MQIA\_TRIGGER\_MSG\_PRIORITY com a chamada MQINQ. Para alterar o valor desse atributo, use a chamada MQSET..

### **TriggerType (MQLONG)**

Isso controla as condições nas quais as mensagens do acionador são gravadas como resultado de mensagens que chegam nesta fila.

Tabela 626. Tipos de fila aos quais este atributo se aplica				
Local	Modelo	Alias	Remoto	Cluster
X	X			

Ele possui um dos seguintes valores:

#### **MQTT\_NONE**

Nenhuma mensagem do acionador é gravada como um resultado de mensagens nesta fila. Isso tem o mesmo efeito da configuração de *TriggerControl* para MQTC\_OFF.

#### **MQTT\_FIRST**

Uma mensagem do acionador é gravada sempre que o número de mensagens de prioridade *TriggerMsgPriority* ou maior na fila for alterado de 0 para 1.

#### **MQTT EVERY**

Uma mensagem do acionador é gravada sempre que uma mensagem de prioridade *TriggerMsgPriority* ou maior chega à fila.

#### **MQTT\_DEPTH**

Uma mensagem do acionador é gravada sempre que o número de mensagens de prioridade *TriggerMsgPriority* ou maior na fila for igual a ou exceder *TriggerDepth*. Após a mensagem do acionador ter sido gravada, *TriggerControl* é configurado como MQTC\_OFF para evitar acionamento adicional até que ele seja explicitamente ativado novamente.

Para determinar o valor desse atributo, use o seletor MQIA\_TRIGGER\_TYPE com a chamada MQINQ.. Para alterar o valor desse atributo, use a chamada MQSET..

### **Uso (MQLONG)**

Isso indica para qual fila é usada..

Tabela 627. Tipos de fila aos quais este atributo se aplica				
Local	Modelo	Alias	Remoto	Cluster
X	X			

O valor é um dos seguintes:

## MQUS\_NORMAL

Esta é uma fila que os aplicativos usam ao colocar e obter mensagens; a fila não é uma fila de transmissão

## MQUS\_TRANSMISSION

Esta é uma fila usada para reter mensagens destinadas a gerenciadores de filas remotas. Quando um aplicativo envia uma mensagem para uma fila remota, o gerenciador de fila local armazena a mensagem temporariamente na fila de transmissão apropriada em um formato especial. Em seguida, um agente do canal de mensagens lê a mensagem da fila de transmissão e transporta a mensagem para o gerenciador de filas remotas. Para obter mais informações sobre como configurar a administração remota, consulte [Configurando gerenciadores de fila para administração remota](#).

Apenas os aplicativos privilegiados podem abrir uma fila de transmissão para MQOO\_OUTPUT para colocar mensagens diretamente nela.. Geralmente, apenas os aplicativos utilitários fazem isso. Assegure-se de que o formato de dados da mensagem esteja correto (consulte "[MQXQH-Cabeçalho da fila de transmissão](#)" na página 641) ou erros podem ocorrer durante o processo de transmissão. O contexto não é transmitido ou configurado a menos que uma das opções de contexto MQPMO\_\*\_CONTEXT seja especificada.

Para determinar o valor desse atributo, use o seletor MQIA\_USAGE com a chamada MQINQ..

## XmitQName (MQCHAR48)

Este é o nome da fila de transmissões. Se esse atributo não estiver em branco quando uma abertura ocorrer, seja para uma fila remota ou para uma definição de alias do gerenciador de fila, ele especificará o nome da fila de transmissão local a ser usada para encaminhar a mensagem.

Tabela 628. Tipos de fila aos quais este atributo se aplica				
Local	Modelo	Alias	Remoto	Cluster
			X	

Se **XmitQName** estiver em branco, a fila local com um nome igual a **RemoteQMgrName** será usada como a fila de transmissão. Se não houver nenhuma fila com o nome **RemoteQMgrName**, a fila identificada pelo atributo do gerenciador de filas **DefXmitQName** será usada.

Este atributo será ignorado se a definição estiver sendo usada como um alias do gerenciador de filas e **RemoteQMgrName** for o nome do gerenciador de fila local. Ele também será ignorado se a definição for usada como uma definição de alias de fila de resposta.

Para determinar o valor desse atributo, use o seletor MQCA\_XMIT\_Q\_NAME com a chamada MQINQ.. O comprimento desse atributo é fornecido por MQ\_Q\_NAME\_LENGTH.

## Atributos para Listas de Nomes

A tabela a seguir resume os atributos específicos para listas de nomes. Os atributos são descritos em ordem alfabética.

Listas de nomes são suportadas em todos os sistemas IBM MQ, mais IBM MQ MQI clients conectados a esses sistemas.

**Nota:** Os nomes dos atributos mostrados nesta seção são nomes descritivos usados com as chamadas MQINQ e MQSET; os nomes são os mesmos dos comandos PCF. Quando comandos MQSC são usados para definir, alterar ou exibir atributos, nomes abreviados alternativos são usados; consulte [Comandos MQSC](#) para obter mais informações.

Tabela 629. Atributos para Listas de Nomes	
Atributo	Descrição
<a href="#">AlterationDate</a>	Data em que a definição foi alterada pela última vez
<a href="#">AlterationTime</a>	Horário em que a definição foi alterada pela última vez
<a href="#">NameCount</a>	Número de nomes na lista de nomes

Tabela 629. Atributos para Listas de Nomes (continuação)

Atributo	Descrição
<u>NamelistDesc</u>	Descrição da lista de nomes.
<u>NamelistName</u>	Nome da lista de nomes
<u>Nomes</u>	Uma lista de nomes <i>NameCount</i>
<u>NamelistType</u>	Tipo de Lista de Nomes
<u>QSGDisp</u>	Disposição do Grupo de Compartilhamento de Fila

### ***AlterationDate (MQCHAR12)***

Esta é a data em que a definição foi mudada pela última vez. O formato da data é YYYY-MM-DD, preenchido com dois espaços em branco finais para tornar o comprimento de 12 bytes.

Para determinar o valor desse atributo, use o seletor MQCA\_ALTERATION\_DATE com a chamadas MQINQ. O comprimento desse atributo é fornecido por MQ\_DATE\_LENGTH

### ***AlterationTime (MQCHAR8)***

Este é o horário em que a definição foi alterada pela última vez. O formato do horário é HH.MM.SS

Para determinar o valor desse atributo, use o seletor MQCA\_ALTERATION\_TIME com a chamada MQINQ.. O comprimento desse atributo é fornecido por MQ\_TIME\_LENGTH

### ***NameCount (MQLONG)***

Esse é o número de nomes na lista de nomes. É maior ou igual a zero. O seguinte valor é definido:

#### **MQNC\_MAX\_NAMELIST\_NAME\_COUNT**

O número máximo de nomes em uma lista de nomes.

Para determinar o valor desse atributo, use o seletor MQIA\_NAME\_COUNT com a chamada MQINQ..

### ***NamelistDesc (MQCHAR64)***

Use este campo para comentário descritivo; seu valor é estabelecido pelo processo de definição. O conteúdo do campo não é significativo para o gerenciador de filas, mas o gerenciador de filas pode requerer que o campo contenha apenas caracteres que possam ser exibidos... Ele não pode conter nenhum caractere nulo; se necessário, será preenchido à direita com espaços em branco. Em uma instalação DBCS, este campo pode conter caracteres DBCS (sujeito a um comprimento máximo de campo de 64 bytes).

**Nota:** Se esse campo contiver caracteres que não estão no conjunto de caracteres do gerenciador de filas (conforme definido pelo atributo do gerenciador de filas **CodedCharSetId**), esses caracteres poderão ser convertidos incorretamente se esse campo for enviado para outro gerenciador de filas..

Para determinar o valor desse atributo, use o seletor MQCA\_NAMELIST\_DESC com a chamada MQINQ..

O comprimento deste atributo é fornecido por MQ\_NAMELIST\_DESC\_LENGTH

### ***NamelistName (MQCHAR48)***

Este é o nome de uma lista de nomes definida no gerenciador de fila local. Para obter mais informações sobre os nomes da lista de nomes, consulte a seção [Outros nomes de objetos](#)

Cada lista de nomes tem um nome diferente dos nomes de outras listas de nomes pertencentes ao gerenciador de filas, mas pode duplicar os nomes de outros objetos do gerenciador de filas de diferentes tipos (por exemplo, filas).

Para determinar o valor desse atributo, use o seletor MQCA\_NAMELIST\_NAME com a chamada MQINQ..

O comprimento desse atributo é fornecido por MQ\_NAMELIST\_NAME\_LENGTH

### ***NamelistType (MQLONG)***



Especifica a natureza dos nomes na lista de nomes e indica como a lista de nomes é usada. Ele é um dos seguintes valores:

#### **MQNT\_NONE**

Lista de nomes sem nenhum tipo designado

#### **MQNT\_Q**

Lista de nomes contendo os nomes de filas.

#### **MQNT\_CLUSTER**

Lista de nomes que contém os nomes de clusters

#### **MQNT\_AUTH\_INFO**

Namelist que contém os nomes de objetos de informações sobre autenticação

Para determinar o valor desse atributo, use o seletor MQIA\_NAMELIST\_TYPE com a chamada MQINQ.

 Esse atributo é suportado apenas no z/OS

### **Nomes (MQCHAR48xNameCount)**

Esta é uma lista de nomes *NameCount*, em que cada nome é o nome de um objeto definido para o gerenciador de filas locais. Para obter mais informações sobre nomes de objetos, consulte [Regras para nomenclatura de objetos IBM MQ](#)

Para determinar o valor desse atributo, use o seletor MQCA\_NAMES com a chamada MQINQ..

O comprimento de cada nome na lista é fornecido por MQ\_OBJECT\_NAME\_LENGTH

### **QSGDisp (MQLONG)**

Isso especifica a disposição da lista de nomes O valor é um dos seguintes:

#### **MQQSGD\_Q\_MGR**


O objeto possui disposição do gerenciador de filas: a definição de objeto é conhecida apenas para o gerenciador de filas locais; a definição não é conhecida para outros gerenciadores de filas no grupo de filas compartilhadas...

Cada gerenciador de filas no grupo de filas compartilhadas pode ter um objeto com o mesmo nome e tipo que o objeto atual, mas eles são objetos separados e não há correlação entre eles Seus atributos não são restritos a serem iguais entre si.

#### **MQQSGD\_COPY**

O objeto é uma cópia local de uma definição de objeto principal que existe no repositório compartilhado. Cada gerenciador de filas no grupo de filas compartilhadas pode ter a sua própria cópia do objeto. Inicialmente, todas as cópias têm os mesmos atributos, mas é possível alterar cada uma delas, usando comandos MQSC, para que seus atributos sejam diferentes daqueles das outras cópias. Os atributos das cópias são ressinchronizados quando a definição principal no repositório compartilhado é alterado..

Para determinar o valor desse atributo, use o seletor MQIA\_QSG\_DISP com a chamada MQINQ.

 Esse atributo é suportado apenas no z/OS

## **Atributos para Definições de Processo**

A tabela a seguir resume os atributos específicos para definições de processo. Os atributos são descritos em ordem alfabética

**Nota:** Os nomes dos atributos nesta seção são nomes descritivos usados com as chamadas MQINQ e MQSET; os nomes são os mesmos que para os comandos PCF Quando comandos MQSC são usados para definir, alterar ou exibir atributos, nomes abreviados alternativos são usados; consulte [Comandos MQSC](#) para obter mais informações.

Tabela 630. Atributos para Definições de Processo

Atributo	Descrição
<u>AlterationDate</u>	Data em que a definição foi alterada pela última vez
<u>AlterationTime</u>	Horário em que a definição foi alterada pela última vez
<u>AppId</u>	Identificador do aplicativo
<u>AppType</u>	Tipo de aplicativo
<u>EnvData</u>	Dados do ambiente
<u>ProcessDesc</u>	Descrição do Processo
<u>ProcessName</u>	Nome do processo
<u>QSGDisp</u>	Disposição do Grupo de Compartilhamento de Fila
<u>UserData</u>	Dados do usuário

### ***AlterationDate (MQCHAR12)***

Esta é a data em que a definição foi mudada pela última vez. O formato da data é YYYY-MM-DD, preenchido com dois espaços em branco finais para tornar o comprimento de 12 bytes.

Para determinar o valor desse atributo, use o seletor MQCA\_ALTERATION\_DATE com a chamadas MQINQ. O comprimento desse atributo é fornecido por MQ\_DATE\_LENGTH

### ***AlterationTime (MQCHAR8)***

Este é o horário em que a definição foi alterada pela última vez. O formato do horário é HH.MM.SS

Para determinar o valor desse atributo, use o seletor MQCA\_ALTERATION\_TIME com a chamada MQINQ. O comprimento desse atributo é fornecido por MQ\_TIME\_LENGTH

### ***AppId (MQCHAR256)***

Esta é uma sequência de caracteres que identifica o aplicativo a ser iniciado. Essas informações são para uso por um aplicativo de monitor acionador que processa mensagens na fila de inicialização; as informações são enviadas à fila de inicialização como parte da mensagem do acionador.

O significado de *AppId* é determinado pelo aplicativo do monitor acionador. O monitor acionador fornecido pelo IBM MQ requer que *AppId* seja o nome de um programa executável. As notas a seguir se aplicam aos ambientes indicados:

- Em z/OS, *AppId* deve ser:
  - Um identificador de transação CICS, para aplicativos iniciados usando a transação CKTI do monitor acionador CICS
  - Um identificador de transação IMS, para aplicativos iniciados usando o IMS monitor acionador CSQQTRMN
- No Windows, o nome do programa pode ser prefixado com um caminho de unidade e de diretório
- No AIX and Linux, o nome do programa pode ser prefixado com um caminho do diretório

A cadeia de caracteres não pode conter nenhum valor nulo. É preenchido à direita com espaços em branco, se necessário.

Para determinar o valor desse atributo, use o seletor MQCA\_APPL\_ID com a chamadas MQINQ. O comprimento desse atributo é fornecido por MQ\_PROCESS\_APPL\_ID\_LENGTH

### ***AppType (MQLONG)***

Isso identifica a natureza do programa a ser iniciado em resposta ao recebimento de uma mensagem do acionador. Essas informações são para uso por um aplicativo de monitor acionador que processa mensagens na fila de inicialização; as informações são enviadas à fila de inicialização como parte da mensagem do acionador.

*ApplType* pode ter qualquer valor, mas os valores a seguir são recomendados para tipos padrão; restrinja os tipos de aplicativos definidos pelo usuário a valores no intervalo MQAT\_USER\_FIRST até MQAT\_USER\_LAST:

**MQAT\_AIX**

AIX (mesmo valor que MQAT\_UNIX).

**MQAT\_BATCH**

aplicativo em lote

**MQAT\_CICS**

Transação CICS .

**MQAT\_IMS**

IMS .

**MQAT\_IMS\_BRIDGE**

aplicativo de ponte IMS .

**MQAT\_JAVA**

Java .

**MQAT\_MVS**

Aplicativo MVS ou TSO (mesmo valor que MQAT\_ZOS).

**MQAT\_OS390**

OS/390 (mesmo valor que MQAT\_ZOS).

**MQAT\_OS400**

IBM i .

**MQAT\_UNIX**

UNIX .

**MQAT\_UNKNOWN**

Aplicação de tipo desconhecido..

**MQAT\_USER**

Aplicativo de usuário

**MQAT\_WINDOWS**

aplicativo Windows de 64 bits.

**MQAT\_WINDOWS\_NT**

aplicativo Windows de 32 bits.

**MQAT\_WLM**

Aplicativo do gerenciador de carga de trabalho do z/OS

**MQAT\_ZOS**

z/OS .

**MQAT\_USER\_FIRST**

Valor mais baixo para o tipo de aplicativo definido pelo usuário.

**MQAT\_USER\_LAST**

Valor mais alto para o tipo de aplicativo definido pelo usuário.

Para determinar o valor desse atributo, use o seletor MQIA\_APPL\_TYPE com a chamada MQINQ.

***EnvData (MQCHAR128)***

Esta é uma sequência de caracteres que contém informações relacionadas ao ambiente pertencentes ao aplicativo a ser iniciado. Essas informações são para uso por um aplicativo de monitor acionador que processa mensagens na fila de inicialização; as informações são enviadas à fila de inicialização como parte da mensagem do acionador.

O significado de *EnvData* é determinado pelo aplicativo do monitor acionador. O monitor acionador fornecido por IBM MQ anexa *EnvData* à lista de parâmetros transmitida para o aplicativo iniciado. A lista

de parâmetros consiste na estrutura MQTMC2 , seguida por um espaço em branco, seguido por *EnvData* com espaços em branco finais removidos. As notas a seguir se aplicam aos ambientes indicados:

- No z/OS:
  - O *EnvData* não é usado pelos aplicativos do monitor acionador fornecidos pelo IBM MQ
  - Se ApplType for MQAT\_WLM, será possível fornecer valores padrão em EnvData para os campos ServiceName e ServiceStep no cabeçalho de informações de trabalho (MQWIH).
- No AIX and Linux, *EnvData* pode ser configurado como o caractere & para executar o aplicativo iniciado no plano de fundo

A cadeia de caracteres não pode conter nenhum valor nulo É preenchido à direita com espaços em branco, se necessário.

Para determinar o valor desse atributo, use o seletor MQCA\_ENV\_DATA com a chamada MQINQ. O comprimento desse atributo é fornecido por MQ\_PROCESS\_ENV\_DATA\_LENGTH

### ***ProcessDesc (MQCHAR64)***

Use este campo para comentário descritivo O conteúdo do campo não é significativo para o gerenciador de filas, mas o gerenciador de filas pode requerer que o campo contenha apenas caracteres que possam ser exibidos... Ele não pode conter nenhum caractere nulo; se necessário, será preenchido à direita com espaços em branco. Em uma instalação DBCS, o campo pode conter caracteres DBCS (sujeito a um comprimento máximo de campo de 64 bytes).

**Nota:** Se esse campo contiver caracteres que não estão no conjunto de caracteres do gerenciador de filas (conforme definido pelo atributo do gerenciador de filas **CodedCharSetId**), esses caracteres poderão ser convertidos incorretamente se esse campo for enviado para outro gerenciador de filas..

Para determinar o valor desse atributo, use o seletor MQCA\_PROCESS\_DESC com a chamada MQINQ.

O comprimento desse atributo é fornecido por MQ\_PROCESS\_DESC\_LENGTH

### ***ProcessName (MQCHAR48)***

Este é o nome de uma definição de processo definida no gerenciador de fila local.

Cada definição de processo tem um nome diferente dos nomes de outras definições de processo pertencentes ao gerenciador de fila. Mas o nome da definição de processo pode ser igual aos nomes de outros objetos do gerenciador de filas de diferentes tipos (por exemplo, filas).

Para determinar o valor desse atributo, use o seletor MQCA\_PROCESS\_NAME com a chamada MQINQ..

O comprimento desse atributo é fornecido por MQ\_PROCESS\_NAME\_LENGTH

### ***QSGDisp (MQLONG)***

Isso especifica a disposição da definição de processo O valor é um dos seguintes:

#### **MQQSGD\_Q\_MGR**


O objeto possui disposição do gerenciador de filas: a definição de objeto é conhecida apenas para o gerenciador de filas locais; a definição não é conhecida para outros gerenciadores de filas no grupo de filas compartilhadas...

Cada gerenciador de filas no grupo de filas compartilhadas pode ter um objeto com o mesmo nome e tipo que o objeto atual, mas eles são objetos separados e não há correlação entre eles Seus atributos não são restritos a serem iguais entre si.

#### **MQQSGD\_COPY**

O objeto é uma cópia local de uma definição de objeto principal que existe no repositório compartilhado. Cada gerenciador de filas no grupo de filas compartilhadas pode ter a sua própria cópia do objeto. Inicialmente, todas as cópias têm os mesmos atributos, mas é possível alterar cada uma delas, usando comandos MQSC, para que seus atributos sejam diferentes daqueles das outras cópias. Os atributos das cópias são resincronizados quando a definição principal no repositório compartilhado é alterado..

Para determinar o valor desse atributo, use o seletor MQIA\_QSG\_DISP com a chamada MQINQ.

 Esse atributo é suportado apenas no z/OS

### **UserData (MQCHAR128)**

*UserData* é uma sequência de caracteres que contém informações sobre o usuário pertencentes ao aplicativo a ser iniciado. Essas informações são para uso por um aplicativo de monitor acionador que processa mensagens na fila de inicialização ou pelo aplicativo que é iniciado pelo monitor acionador. As informações são enviadas à fila de inicialização como parte da mensagem do acionador.

O significado de *UserData* é determinado pelo aplicativo do monitor acionador. O monitor acionador fornecido por IBM MQ passa *UserData* para o aplicativo iniciado como parte da lista de parâmetros. A lista de parâmetros consiste na estrutura MQTMC2 (contendo *UserData*), seguida por um espaço em branco, seguido por *EnvData* com espaços em branco finais removidos.

A cadeia de caracteres não pode conter nenhum valor nulo. É preenchido à direita com espaços em branco, se necessário. Para o Microsoft Windows, a sequência de caractere não deverá conter aspas duplas se a definição de processo for transmitida para **runmqtrm**.

Para determinar o valor desse atributo, use o seletor MQCA\_USER\_DATA com a chamada MQINQ.. O comprimento desse atributo é fornecido por MQ\_PROCESS\_USER\_DATA\_LENGTH.

## **Códigos de retorno**

Para cada chamada de IBM MQ Message Queue Interface (MQI) e IBM MQ Administration Interface (MQAI), um código de **conclusão** e um código de **razão** são retornados pelo gerenciador de filas ou por uma rotina de saída, para indicar o sucesso ou falha da chamada.

Os aplicativos não devem depender de erros que estejam sendo verificados em uma ordem específica, exceto quando especificamente indicado. Se mais de um código de conclusão ou código de razão poderia surgir de uma chamada, o erro específico relatado depende da implementação.

Os aplicativos que verificam a conclusão bem-sucedida após uma chamada API IBM MQ devem sempre verificar o código de conclusão. Não assumam o valor do código de conclusão com base no valor do código de razão.

### **Códigos de conclusão**

O parâmetro do código de conclusão (*CompCode*) permite que o responsável pela chamada veja rapidamente se a chamada foi concluída com êxito, parcialmente ou falhou. A seguir está uma lista de códigos de conclusão, com mais detalhes do que é fornecido nas descrições de chamada:

#### **MQCC\_OK**

A chamada foi concluída completamente; todos os parâmetros de saída foram definidos. O parâmetro **Reason** sempre tem o valor MQRC\_NONE nesse caso.

#### **MQCC\_WARNING**

A chamada foi parcialmente concluída. Alguns parâmetros de saída foram definidos além dos parâmetros de saída *CompCode* e *Reason*. O parâmetro **Reason** fornece informações adicionais sobre a conclusão parcial.

#### **MQCC\_FAILED**

O processamento da chamada não foi concluído. O estado do gerenciador de filas é inalterado. Os parâmetros de saída *CompCode* e *Reason* foram configurados; outros parâmetros permanecem inalterados.

A razão pode ser uma falha no programa aplicativo ou pode ser o resultado de alguma situação externa ao programa, por exemplo, a autoridade do usuário pode ter sido revogada. O parâmetro **Reason** fornece informações adicionais sobre o erro..

## Códigos de Razão

O parâmetro de código de razão (*Reason*) qualifica o parâmetro de código de conclusão (*CompCode*)..

Se não houver razão especial a ser relatada, MQRC\_NONE será retornado. Uma chamada bem-sucedida retorna MQCC\_OK e MQRC\_NONE.

Se o código de conclusão for MQCC\_WARNING ou MQCC\_FAILED, o gerenciador de filas sempre relatará uma razão de qualificação; os detalhes serão fornecidos em cada descrição da chamada.

Onde as rotinas de saída de usuário configuram códigos de conclusão e motivos, elas devem aderir a essas regras. Além disso, quaisquer valores de razão especiais definidos por saídas de usuário devem ser menores que zero, para assegurar que não entrem em conflito com valores definidos pelo gerenciador de filas. As saídas podem configurar motivos já definidos pelo gerenciador de filas, onde apropriado

Códigos de razão também ocorrem em:

- O campo *Reason* da estrutura MQDLH
- O campo *Feedback* da estrutura MQMD

Para obter descrições completas de códigos de razão, consulte [Mensagens e códigos de razão](#).

## Regras para validar opções de MQI

Esta seção lista as situações que produzem um código de razão MQRC\_OPTIONS\_ERROR de uma chamada MQOPEN, MQPUT, MQPUT1, MQGET, MQCLOSE ou MQSUB.

### chamada MQOPEN

Para as opções da chamada MQOPEN:

- Pelo menos *um* dos seguintes deve ser especificado:

- MQOO\_BROWSE
- MQOO\_INPUT\_EXCLUSIVE <sup>1</sup>
- MQOO\_INPUT\_SHARED <sup>1</sup>
- MQOO\_INPUT\_AS\_Q\_DEF <sup>1</sup>
- MQOO\_INQUIRE
- MQOO\_OUTPUT
- MQOO\_SET
- MQOO\_BIND\_ON\_OPEN <sup>2</sup>
- MQOO\_BIND\_NOT\_FIXED <sup>2</sup>
- MQOO\_BIND\_ON\_GROUP <sup>2</sup>
- MQOO\_BIND\_AS\_Q\_DEF <sup>2</sup>

- Apenas *um* dos seguintes é permitido:

- MQOO\_READ\_AHEAD
- MQOO\_NO\_READ\_AHEAD
- MQOO\_READ\_AHEAD\_AS\_Q\_DEF

1. Apenas *um* dos seguintes é permitido:

- MQOO\_INPUT\_EXCLUSIVE
- MQOO\_INPUT\_SHARED
- MQOO\_INPUT\_AS\_Q\_DEF

2. Apenas *um* dos seguintes é permitido:

- MQOO\_BIND\_ON\_OPEN

- MQOO\_BIND\_NOT\_FIXED
- MQOO\_BIND\_ON\_GROUP
- MQOO\_BIND\_AS\_Q\_DEF

**Nota:** As opções listadas anteriormente são mutuamente exclusivas. No entanto, como o valor de MQOO\_BIND\_AS\_Q\_DEF é zero, especificá-lo com qualquer uma das outras duas opções de ligação não resulta no código de razão MQRC\_OPTIONS\_ERROR MQOO\_BIND\_AS\_Q\_DEF é fornecido para a documentação do programa de auxílio

- Se MQOO\_SAVE\_ALL\_CONTEXT for especificado, uma das opções MQOO\_INPUT\_ \* também deverá ser especificada.
- Se uma das opções MQOO\_SET\_ \* \_CONTEXT ou MQOO\_PASS\_ \* \_CONTEXT for especificada, MQOO\_OUTPUT também deverá ser especificado.
- Se MQOO\_CO\_OP for especificado, MQOO\_BROWSE também deverá ser especificado
- Se MQOO\_NO\_MULTICAST for especificado, MQOO\_OUTPUT também deverá ser especificado

## Chamada MQPUT

Para as opções put-message:

- A combinação de MQPMO\_SYNCPOINT e MQPMO\_NO\_SYNCPOINT não é permitida..
- Apenas *um* dos seguintes é permitido:
  - MQPMO\_DEFAULT\_CONTEXT
  - MQPMO\_NO\_CONTEXT
  - MQPMO\_PASS\_ALL\_CONTEXT
  - MQPMO\_PASS\_IDENTITY\_CONTEXT
  - MQPMO\_SET\_ALL\_CONTEXT
  - MQPMO\_SET\_IDENTITY\_CONTEXT
- Apenas *um* dos seguintes é permitido:
  - MQPMO\_ASYNC\_RESPONSE
  - MQPMO\_SYNC\_RESPONSE
  - MQPMO\_RESPONSE\_AS\_TOPIC\_DEF
  - MQPMO\_RESPONSE\_AS\_Q\_DEF
- MQPMO\_ALTERNATE\_USER\_AUTHORITY não é permitido (é válido apenas na chamada MQPUT1)

## chamada MQPUT1

Para as opções put-message, as regras são as mesmas para a chamada MQPUT, exceto para o seguinte:

- MQPMO\_ALTERNATE\_USER\_AUTHORITY é permitido.
- MQPMO\_LOGICAL\_ORDER não é permitido..

## Chamada MQGET

Para as opções get-message:

- Apenas *um* dos seguintes é permitido:
  - MQGMO\_NO\_SYNCPOINT
  - MQGMO\_SYNCPOINT
  - MQGMO\_SYNCPOINT\_IF\_PERSISTENT
- Apenas *um* dos seguintes é permitido:

- MQGMO\_BROWSE\_FIRST
- MQGMO\_BROWSE\_MSG\_UNDER\_CURSOR
- MQGMO\_BROWSE\_NEXT
- MQGMO\_MSG\_UNDER\_CURSOR
- MQGMO\_SYNCPOINT não é permitido com nenhum dos seguintes:
  - MQGMO\_BROWSE\_FIRST
  - MQGMO\_BROWSE\_MSG\_UNDER\_CURSOR
  - MQGMO\_BROWSE\_NEXT
  - MQGMO\_LOCK
  - MQGMO\_UNLOCK
- MQGMO\_SYNCPOINT\_IF\_PERSISTENT não é permitido com nenhum dos seguintes:
  - MQGMO\_BROWSE\_FIRST
  - MQGMO\_BROWSE\_MSG\_UNDER\_CURSOR
  - MQGMO\_BROWSE\_NEXT
  - MQGMO\_COMPLETE\_MSG
  - MQGMO\_UNLOCK
- MQGMO\_MARK\_SKIP\_BACKOUT requer que MQGMO\_SYNCPOINT seja especificado.
- A combinação de MQGMO\_WAIT e MQGMO\_SET\_SIGNAL não é permitida
- Se MQGMO\_LOCK for especificado, um dos seguintes também deverá ser especificado:
  - MQGMO\_BROWSE\_FIRST
  - MQGMO\_BROWSE\_MSG\_UNDER\_CURSOR
  - MQGMO\_BROWSE\_NEXT
- Se MQGMO\_UNLOCK for especificado, somente os seguintes valores serão permitidos:
  - MQGMO\_NO\_SYNCPOINT
  - MQGMO\_NO\_WAIT

### **chamada MQCLOSE**

Para as opções da chamada MQCLOSE:

- A combinação de MQCO\_DELETE e MQCO\_DELETE\_PURGE não é permitida
- Apenas um dos seguintes é permitido:
  - MQCO\_KEEP\_SUB
  - MQCO\_REMOVE\_SUB

### **Chamada MQSUB**

Para as opções da chamada MQSUB:

- Pelo menos um dos seguintes deve ser especificado:
  - MQSO\_ALTER
  - MQSO\_RESUME
  - MQSO\_CREATE
- Apenas um dos seguintes é permitido:
  - MQSO\_DURABLE
  - MQSO\_NON\_DURABLE



**Nota:** As opções listadas anteriormente são mutuamente exclusivas. No entanto, como o valor de MQSO\_NON\_DURABLE é zero, especificá-lo com MQSO\_DURABLE não resulta em código de razão MQRC\_OPTIONS\_ERROR. MQSO\_NON\_DURABLE é fornecido para a documentação do programa de auxílio

- A combinação de MQSO\_GROUP\_SUB e MQSO\_MANAGED não é permitida
- MQSO\_GROUP\_SUB requer que MQSO\_SET\_CORREL\_ID seja especificado.
- Apenas um dos seguintes é permitido:
  - MQSO\_ANY\_USERID
  - MQSO\_FIXED\_USERID
- MQSO\_NEW\_PUBLICATIONS\_ONLY é permitido em combinação com:
  - MQSO\_CREATE
  - MQSO\_ALTER, se MQSO\_NEW\_PUBLICATIONS\_ONLY foi configurado na assinatura original
- A combinação de MQSO\_PUBLICATIONS\_ON\_REQUEST e SubLevel maior que 1 não é permitida.
- Apenas um dos seguintes é permitido:
  - MQSO\_WILDCARD\_CHAR
  - MQSO\_WILDCARD\_TOPIC
- MQSO\_NO\_MULTICAST requer que MQSO\_MANAGED seja especificado.

## Mensagens de comando de publicação / assinatura enfileirados

Um aplicativo pode usar mensagens de comandos do MQRFH2 para controlar um aplicativo de publicação / assinatura enfileirada

Um aplicativo que está usando o MQRFH2 para publicar / assinar pode enviar as mensagens de comandos a seguir para o SYSTEM.SYSTEM.BROKER.CONTROL.QUEUE:

- [“Mensagem Delete Publication” na página 910](#)
- [“Remover Registro da Mensagem do Assinante” na página 911](#)
- [“Publicar mensagem” na página 915](#)
- [“Mensagem Registrar Assinante” na página 917](#)
- [“Mensagem de atualização de solicitação” na página 922](#)

Se você estiver gravando aplicativos de publicação / assinatura enfileirados, deverá entender essas mensagens, a mensagem de resposta do gerenciador de filas e o descritor de mensagens (MQMD); consulte as informações a seguir:

- [“Mensagem de Resposta do Gerenciador de Filas” na página 924](#)
- [“Configurações do MQMD para publicações encaminhadas por um gerenciador de filas” na página 930](#)
- [“Configurações do MQMD em mensagens de resposta do gerenciador de filas” na página 931](#)
- [“Códigos de Razão de Publicação / Assinatura” na página 926](#)

Os comandos estão contidos em uma pasta psc no campo **NameValueData** do cabeçalho MQRFH2. A mensagem que pode ser enviada por um broker em resposta a uma mensagem de comando está contida em uma pasta pscr

As descrições de cada comando listam as propriedades que podem estar contidas em uma pasta. A menos que seja especificado de outra forma, as propriedades são opcionais e podem ocorrer apenas uma vez

Os nomes de propriedades são mostrados como <Command>

Os valores devem estar no formato de sequência, por exemplo: Publish

Uma constante de sequência que representa o valor de uma propriedade é mostrada entre parênteses, por exemplo: (MQPSC\_PUBLISH).

Constantes de sequência são definidas no arquivo de cabeçalho cmqpsc . h fornecido com o gerenciador de filas.

## Mensagem Delete Publication

A mensagem de comando **Delete Publication** é enviada a um gerenciador de fila de um publicador ou de outro gerenciador de filas para informar ao gerenciador de filas para excluir quaisquer publicações retidas para os tópicos especificados.

Essa mensagem é enviada para uma fila monitorada por interface de publicação / assinatura enfileirada do gerenciador de filas.

A fila de entrada deve ser a fila para a qual a publicação original foi enviada.

Se possuir autoridade para alguns, mas não para todos os tópicos que são especificados na mensagem de comando **Delete Publication**, somente aqueles tópicos serão excluídos. Uma mensagem **Broker Response** indica quais tópicos não foram excluídos.

De forma semelhante, se um comando **Publish** contiver mais de um tópico, um comando **Delete Publication** que corresponda a alguns desses tópicos, mas não a todos eles, excluirá somente as publicações para os tópicos que são especificados no comando **Delete Publication**.

Consulte [“Configurações do MQMD para publicações encaminhadas por um gerenciador de filas” na página 930](#) para obter detalhes dos parâmetros do descritor de mensagens (MQMD) que são necessários ao enviar uma mensagem de comando para o gerenciador de filas.

## Propriedades

### Comando (MQPSC\_COMMAND)

O valor é DeletePub (MQPSC\_DELETE\_PUBLICATION).

Essa propriedade precisa ser especificada.

### Tópico> (MQPSC\_TOPIC)

O valor é uma sequência que contém um tópico para o qual as publicações retidas devem ser excluídas. Caracteres curinga podem ser incluídos na sequência para excluir publicações em mais de um tópico.

Essa propriedade precisa ser especificada; ela pode ser repetida para tantos tópicos quanto necessário.

### DelOpt (MQPSC\_DELETE\_OPTION)

A propriedade delete options pode ter um dos seguintes valores:

#### Local (MQPSC\_LOCAL)

Todas as publicações retidas para os tópicos especificados são excluídas no gerenciador de fila local (ou seja, o gerenciador de filas para o qual essa mensagem é enviada), se elas foram publicadas com a opção Local ou não...

As publicações em outros gerenciadores de filas não são afetados

#### Nenhum (MQPSC\_NONE)

Todas as opções assumem seus valores padrão. Isso tem o mesmo efeito que omitir a propriedade DelOpt. Se outras opções forem especificadas ao mesmo tempo, Nenhuma será ignorada.

O padrão se essa propriedade for omitida é que todas as publicações retidas para os tópicos especificados sejam excluídas em todos os gerenciadores de filas na rede, independentemente de terem sido publicadas com a opção Local .

## exemplo

Exemplo de NameValueData para uma mensagem de comando **Delete Publication**. Isso é usado pelo aplicativo de amostra para excluir, no gerenciador da fila local, a publicação retida que contém a pontuação mais recente na correspondência entre Team1 e Team2.

```
<psc>
  <Command>DeletePub</Command>
  <Topic>Sport/Soccer/State/LatestScore/Team1 Team2</Topic>
  <DelOpt>Local</DelOpt>
</psc>
```

## Remover Registro da Mensagem do Assinante

A mensagem de comando **Deregister Subscriber** é enviada para um gerenciador de fila por um assinante, ou por outro aplicativo em nome de um assinante, para indicar que ele não deseja mais receber mensagens correspondentes aos parâmetros fornecidos

Esta mensagem é enviada para o SYSTEM.BROKER.CONTROL.QUEUE, a fila de controle do gerenciador de filas. O usuário deve ter autoridade necessária para colocar uma mensagem nesta fila.

Consulte [Configurações de MQMD para publicações encaminhadas por um gerenciador de filas](#) para obter detalhes dos parâmetros do descritor de mensagens (MQMD) necessários ao enviar uma mensagem de comando para o gerenciador de filas.

Uma assinatura individual pode ter o registro removido especificando os valores de tópico, ponto de assinatura e filtro correspondentes da assinatura original. Se algum dos valores não tiver sido especificado (ou seja, eles tomaram os valores padrão) na assinatura original, eles deverão ser omitidos quando a assinatura tiver o registro removido.

Todas as assinaturas para um assinante ou um grupo de assinantes podem ter o registro removido usando a opção `DeregAll`. Por exemplo, se `DeregAll` for especificado, juntamente com um ponto de assinatura (mas nenhum tópico ou filtro), todas as assinaturas para o assinante no ponto de assinatura especificado serão removidas, independentemente do tópico e do filtro. Qualquer combinação de tópico, filtro e ponto de assinatura é permitida; se todos os três forem especificados, apenas uma assinatura poderá corresponder e a opção `DeregAll` será ignorada.

A mensagem deve ser enviada pelo assinante que registrou a assinatura; isso é confirmado verificando o ID do usuário do assinante.

As assinaturas também podem ser removidas por um administrador do sistema usando comandos MQSC ou PCF. No entanto, as assinaturas registradas com uma fila dinâmica temporária são associadas com a fila, não apenas o nome da fila. Se a fila for excluída, explicitamente ou pelo aplicativo desconectando do gerenciador de filas, não será mais possível usar o comando **Deregister Subscriber** para remover o registro das assinaturas para essa fila. As assinaturas podem ter o registro removido usando o ambiente de trabalho do desenvolvedor e serão removidas automaticamente pelo gerenciador de filas na próxima vez que ele corresponder a uma publicação para a assinatura ou na próxima vez que o gerenciador de filas for reiniciado. Em circunstâncias normais, os aplicativos devem cancelar o registro de assinaturas antes de excluir a fila ou desconectar do gerenciador de filas..

Se um assinante enviar uma mensagem para remover o registro de uma assinatura e receber uma mensagem de resposta para dizer que ela foi processada com êxito, algumas publicações ainda poderão atingir a fila de assinantes se estiverem sendo processadas pelo gerenciador de filas ao mesmo tempo em que a assinatura estiver sendo removida do registro. Se as mensagens não forem removidas da fila, poderá haver uma acumulação de mensagens não processadas na fila de assinantes. Se o aplicativo executar um loop que inclui uma chamada MQGET com o `CorrelId` apropriado após a suspensão por um tempo, essas mensagens serão removidas da fila

Da mesma forma, se o assinante usar uma fila dinâmica permanente, e cancelar o registro e fechar a fila com a opção `MQCO_DELETE_PURGE` em uma chamada MQCLOSE, a fila poderá não estar vazia. Se quaisquer publicações do gerenciador de filas ainda não forem confirmadas quando a fila for excluída, um

código de retorno MQRC\_Q\_NOT\_EMPTY será emitido pela chamada MQCLOSE. O aplicativo pode evitar esse problema dormindo e emitindo novamente a chamada MQCLOSE de tempos em tempos..

## **Propriedades**

### **Comando (MQPSC\_COMMAND)**

O valor é DeregSub (MQPSC\_DEREGISTER\_ASSINANTE).

Essa propriedade precisa ser especificada.

### **Tópico (MQPSC\_TOPIC)**

O valor é uma sequência que contém o tópico cujo registro será removido.

Essa propriedade pode, opcionalmente, ser repetida se vários tópicos tiverem o registro removido. Ele poderá ser omitido se DeregAll for especificado em <RegOpt>

Os tópicos especificados podem ser um subconjunto daqueles que são registrados se o assinante desejar reter assinaturas para outros tópicos. Caracteres curinga são permitidos, mas uma sequência de tópicos que contém caracteres curinga deve corresponder exatamente à sequência correspondente que foi especificada na mensagem de comando **Deregister Subscriber**.

### **SubPoint (MQPSC\_SUBSCRIPTION\_POINT)**

O valor é uma sequência que especifica o ponto de assinatura a partir do qual a assinatura deve ser separada.

Essa propriedade não deve ser repetida Ele poderá ser omitido se um < Topic> for especificado ou se o DeregAll for especificado em <RegOpt> Se você omitir essa propriedade, acontecerá o seguinte:

- Se você **não** especificar DeregAll, as assinaturas correspondentes à propriedade < Topic> (e a propriedade < Filter> , se presente) serão removidas do registro do ponto de assinatura padrão.
- Se você especificar DeregAll, todas as assinaturas (correspondentes às propriedades < Topic> e < Filter> se presentes) serão removidas do registro de todos os pontos de assinatura.

Observe que não é possível especificar o ponto de assinatura padrão explicitamente. Portanto, não há nenhuma maneira de remover o registro de todas as assinaturas desse ponto de assinatura apenas; você deve especificar os tópicos.

### **SubIdentity (MQPSC\_SUBSCRIPTION\_IDENTITY)**

Esta é uma sequência de comprimento variável com um comprimento máximo de 64 caracteres.. Ele é usado para representar um aplicativo com interesse em uma assinatura. O gerenciador de filas mantém um conjunto de identidades de assinante para cada subscrição Cada assinatura pode permitir que seu conjunto de identidade contenha apenas uma única identidade ou um número ilimitado de identidades..

Se o SubIdentity estiver no conjunto de identidade para a assinatura, ele será removido do conjunto. Se o conjunto de identidade ficar vazio como resultado disso, a assinatura será removida do gerenciador de filas, a menos que LeaveOnly seja especificado como um valor da propriedade RegOpt . Se o conjunto de identidades ainda contiver outras identidades, a assinatura não será removida do gerenciador de filas e o fluxo de publicação não será interrompido

Se SubIdentity for especificado, mas o SubIdentity não estiver no conjunto de identidade para a assinatura, então o comando **Deregister Subscriber** falhará com o código de retorno MQRCCF\_SUB\_IDENTITY\_ERROR

### **Filtro (MQPSC\_FILTER)**

O valor é uma sequência que especifica o filtro cujo registro será removido. Ele deve corresponder exatamente, incluindo maiúsculas e minúsculas e espaços, um filtro de assinatura que foi registrado anteriormente.

Essa propriedade pode, opcionalmente, ser repetida se mais de um filtro tiver o registro cancelado. Ele poderá ser omitido se um < Topic> for especificado ou se o DeregAll for especificado em <RegOpt>

Os filtros especificados podem ser um subconjunto daqueles registrados se o assinante desejar reter assinaturas para outros filtros

### **RegOpt (MQPSC\_REGISTRATION\_OPTION)**

A propriedade de opções de registro pode ter os seguintes valores:

#### **DeregAll**

(MQPSC\_DEREGISTER\_ALL)

Todas as assinaturas correspondentes registradas para este assinante devem ter o registro removido.

Se você especificar DeregAll:

- < Topic>, <SubPoint> e < Filter> podem ser omitidos..
- < Topic> e < Filter> podem ser repetidos, se necessário.
- <SubPoint> não deve ser repetido

Se você **não** especificar DeregAll:

- < Topic> deve ser especificado e pode ser repetido, se necessário,
- <SubPoint> e < Filtro> podem ser omitidos
- <SubPoint> não deve ser repetido
- < Filtro> pode ser repetido, se necessário

Se os tópicos e filtros forem repetidos, todas as assinaturas correspondentes a todas as combinações das duas serão removidas. Por exemplo, um comando **Deregister Subscriber** que especifica três tópicos e três filtros tentará remover nove assinaturas..

#### **CorrelAsId**

(MQPSC\_CORREL\_ID\_AS\_IDENTITY)

O CorrelId no descritor de mensagens (MQMD), que não deve ser zero, é usado para identificar o assinante. Ele deve corresponder ao CorrelId usado na assinatura original

#### **FullResp**

(MQPSC\_FULL\_RESPONSE)

Quando FullResp for especificado, todos os atributos da assinatura serão retornados na mensagem de resposta, se o comando não falhar..

Quando FullResp é especificado, DeregAll não é permitido no comando **Deregister Subscriber** Também não é possível especificar vários tópicos. O comando falha com o código de retorno *MQRCCF\_REG\_OPTIONS\_ERROR* em ambos os casos.

#### **LeaveOnly**

(MQPSC\_LEAVE\_ONLY)

Quando você especifica isso com um SubIdentity que está no conjunto de identidade para a assinatura, o SubIdentity é removido do conjunto de identidade para a assinatura. A assinatura não é removida do gerenciador de filas, mesmo se o conjunto de identidade resultante estiver vazio. Se o valor SubIdentity não estiver na identidade configurada, o comando falhará com o código de retorno *MQRCCF\_SUB\_IDENTITY\_ERROR*.

Se LeaveOnly for especificado sem SubIdentity, o comando falhará com código de retorno *MQRCCF\_REG\_OPTIONS\_ERROR*.

Se nem LeaveOnly nem um SubIdentity forem especificados, a assinatura será removida, independentemente do conteúdo do conjunto de identidade para a assinatura

#### **Nenhum**

(MQPSC\_NONE)

Todas as opções assumem seus valores padrão. Isso tem o mesmo efeito que omitir a propriedade de opções de registro. Se outras opções forem especificadas ao mesmo tempo, nenhuma será ignorada.

#### **ID doVariableUser**

(MQPSC\_VARIABLE\_USER\_ID)

Quando especificado, a identidade do assinante (fila, gerenciador de filas e correlid) não é restrita a um único ID do usuário... Isso difere do comportamento existente do gerenciador de filas que associa o ID do usuário da mensagem de registro original à identidade do assinante e, a partir daí, evita que qualquer outro usuário use essa identidade. Se um novo assinante tentar usar a mesma identidade o código de retorno *MQRCCF\_DUPLICATE\_SUBSCRIPTION* será retornado.

Qualquer usuário pode modificar ou cancelar o registro da assinatura quando tiver a autoridade adequada, evitando a verificação existente de que o ID do usuário deve corresponder ao do assinante original.

Para incluir essa opção em uma assinatura existente, o comando deve vir do mesmo ID do usuário da própria assinatura original.

Se a assinatura a ser removida do registro tiver o `VariableUserId` configurado, ele deverá ser configurado no momento do cancelamento do registro para indicar qual assinatura está sendo removida do registro. Caso contrário, o ID do usuário do comando **Deregister Subscriber** será usado para identificar a subscrição. Isso será substituído, juntamente com os outros identificadores de assinante, se um nome de assinatura for fornecido.

O padrão, se essa propriedade for omitida, é que nenhuma opção de registro está configurada.

#### **QMgrName (MQPSC\_Q\_MGR\_NAME)**

O valor é o nome do gerenciador de filas para a fila de assinantes. Ela deve corresponder ao `QMgrName` usado na assinatura original.

Se essa propriedade for omitida, o padrão é o nome de `ReplyToQMgr` no descritor da mensagem (MQMD). Se o nome resultante estiver em branco, ele será padronizado com o nome do gerenciador de filas.

#### **QName (MQPSC\_Q\_NAME)**

O valor é o nome da fila de assinante. Ele deve corresponder ao `QName` utilizado na assinatura original.

Se essa propriedade for omitida, o padrão será o nome `ReplyToQ` no descritor de mensagem (MQMD), que não deve estar em branco.

#### **SubName (MQPSC\_SUBSCRIPTION\_NAME)**

Se você especificar `SubName` em um comando **Deregister Subscriber**, o valor `SubName` terá precedência sobre todos os outros campos de identificador, exceto o ID do usuário, a menos que o `VariableUserId` seja configurado na própria assinatura. Se o `VariableUserId` não for configurado, o comando **Deregister Subscriber** será bem-sucedido somente se o ID do usuário da mensagem de comando corresponder ao da assinatura, se não, o comando falhará com o código de retorno *MQRCCF\_DUPLICATE\_IDENTITY*.

Se existir uma assinatura que corresponda à identidade tradicional desse comando, mas não tiver `SubName` o comando **Deregister Subscriber** falhará com o código de retorno *MQRCCF\_SUB\_NAME\_ERROR*. Se for feita uma tentativa de remover o registro de uma assinatura que tenha um `SubName` usando uma mensagem de comando que corresponda à identidade tradicional, mas sem `SubName` especificado, o comando será bem-sucedido.

#### **SubUserDados (MQPSC\_SUBSCRIPTION\_USER\_DATA)**

Esta é uma cadeia de texto de comprimento variável. O valor é armazenado pelo gerenciador de filas com a assinatura, mas não tem influência na entrega da publicação para o assinante. O valor pode ser alterado registrando novamente na mesma assinatura com um novo valor. Este atributo é para uso do aplicativo.

SubUserOs dados são retornados nas informações de Metatópico (*MQCACF\_REG\_SUB\_USER\_DATA*) para uma assinatura, se `SubUserDados` estiverem presentes.

## exemplo

Aqui está um exemplo de `NameValueData` para uma mensagem de comando **Deregister Subscriber** .. Neste exemplo, o aplicativo de amostra está removendo o registro de sua assinatura para os tópicos que contêm a última pontuação para todas as correspondências. A identidade do assinante, incluindo o `CorrelId`, é obtida dos padrões no MQMD.

```
<psc>
  <Command>DeregSub</Command>
  <RegOpt>CorrelAsId</RegOpt>
  <Topic>Sport/Soccer/State/LatestScore/#</Topic>
</psc>
```

## Publicar mensagem

A mensagem de comando **Publish** é colocada em uma fila, ou de um gerenciador de filas para um assinante, para publicar informações sobre um tópico ou tópicos especificados

É necessária autoridade para colocar uma mensagem em uma fila e autoridade para publicar informações sobre um tópico ou tópicos especificados.

Se o usuário tiver autoridade para publicar informações em alguns, mas não em todos, tópicos, apenas esses tópicos serão usados para publicar; uma resposta de aviso indica quais tópicos não serão usados para publicar

Se um assinante tiver quaisquer assinaturas correspondentes, o gerenciador de filas encaminhará a mensagem **Publish** para as filas de assinantes definidas nas mensagens de comandos do **Register Subscriber** correspondentes

Consulte [Mensagem de resposta do gerenciador de filas](#) para obter detalhes dos parâmetros do descritor de mensagens (MQMD) necessários ao enviar uma mensagem de comando para o gerenciador de filas e usado quando um gerenciador de filas encaminha uma publicação para um assinante.

O gerenciador de filas encaminha a mensagem **Publish** para outros gerenciadores de fila na rede que possuem assinaturas correspondentes, a menos que seja uma publicação local..

Os dados da publicação, se existirem, são incluídos no corpo da mensagem. Os dados podem ser descritos em uma pasta `<mcd>` no campo `NameValueData` do cabeçalho `MQRFH2` .

## Propriedades

### Comando (*MQPSC\_COMMAND*)

O valor é `Publish` (*MQPSC\_PUBLISH*)

Essa propriedade precisa ser especificada.

### Tópico (*MQPSC\_TOPIC*)

O valor é uma sequência que contém um tópico que categoriza essa publicação. Nenhum caractere curinga é permitido.

Você deve incluir o tópico na lista de nomes `SYSTEM.QPUBSUB.QUEUE.NAMELIST`, consulte [Incluindo um Fluxo](#) para obter instruções sobre como concluir esta tarefa

Essa propriedade precisa ser especificada e pode ser repetida opcionalmente para tantos tópicos quanto necessário.

### SubPoint (*MQPSC\_SUBSCRIPTION\_POINT*).

O ponto de assinatura no qual a publicação é publicada.

Em WebSphere Event Broker 6.0, o valor da propriedade `<SubPoint>` é o valor do atributo de Ponto de assinatura do nó de Publicação que está manipulando a publicação.

No IBM WebSphere MQ 7.0.1, o valor da propriedade `<SubPoint>` deve corresponder ao nome de um ponto de assinatura. Consulte [Incluindo um ponto de assinatura](#)..

### **PubOpt (MQPSC\_PUBLICATION\_OPTION)**

A propriedade publication options pode ter os seguintes valores:

#### **RetainPub**

(MQPSC\_RETAIN\_PUB)

O gerenciador de filas deve manter uma cópia da publicação. Se essa opção não for configurada, a publicação será excluída assim que o gerenciador de filas tiver enviado a publicação para todos os seus assinantes atuais..

#### **IsRetainedPub**

(MQPSC\_IS\_RETAINED\_PUB)

(Pode ser configurado apenas por um gerenciador de filas.). Esta publicação foi mantida pelo gerenciador de fila. O gerenciador de filas configura essa opção para notificar um assinante de que essa publicação foi publicada anteriormente e retida, desde que a assinatura tenha sido registrada com a opção InformIfRetido . Ela é definida somente em resposta a uma mensagem de comando Registrar Assinante ou Pedido de Atualização. Publicações retidas que são enviadas diretamente a assinantes não têm essa opção definida.

#### **Local**

(MQPSC\_LOCAL)

Esta opção informa ao gerenciador de filas que esta publicação não deve ser enviada para outros gerenciadores de filas Todos os assinantes que se registraram neste gerenciador de filas receberão esta publicação, se tiverem assinaturas correspondentes

#### **OtherSubsOnly**

(MQPSC\_OTHER\_SUBS\_ONLY)

Esta opção permite um processamento mais simples de aplicativos tipo conferência, em que um publicador também é um assinante do mesmo tópico. Ele informa ao gerenciador de filas para não enviar a publicação para a fila de assinantes do publicador mesmo se ele tiver uma assinatura correspondente. A fila de assinantes do publicador consiste em seu QMgrName, QNamee opcional CorrelId, conforme descrito na lista a seguir.

#### **CorrelAsId**

(MQPSC\_CORREL\_ID\_AS\_IDENTITY)

O CorrelId no MQMD (que não pode ser zero) faz parte da fila de assinante do publicador em aplicativos nos quais o publicador também é um assinante.

#### **Nenhum**

(MQPSC\_NONE)

Todas as opções assumem seus valores padrão. Isso tem o mesmo efeito que omitir a propriedade publication options. Se outras opções forem especificadas ao mesmo tempo, Nenhuma será ignorada.

É possível ter mais de uma opção de publicação, introduzindo elementos <PubOpt> adicionais

O padrão, se essa propriedade for omitida, é que nenhuma opção de publicação é definida.

### **PubTime (MQPSC\_PUBLISH\_TIMESTAMP)**

O valor é a data e hora da publicação opcional, definida pelo publicador. Tem 16 caracteres de comprimento com o formato:

```
YYYYMMDDHHMSSSTH
```

utilizando Hora Universal. Essas informações não são verificadas pelo gerenciador de filas antes de serem enviadas para os assinantes

### **SeqNum (MQPSC\_SEQUENCE\_NUMBER)**

O valor é um número de sequência opcional definido pelo publicador.



Ele deve ser incrementado em 1 com cada publicação. No entanto, isso não é verificado pelo gerenciador de filas, que apenas transmite essas informações aos assinantes..

Se publicações no mesmo tópico forem publicadas em gerenciadores de filas interconectados diferentes, é responsabilidade dos publicadores assegurar que os números de sequência, se usados, sejam significativos.

#### **QMgrName (MQPSC\_Q\_MGR\_NAME)**

O valor é uma sequência que contém o nome do gerenciador de filas para a fila de assinantes do publicador, em aplicativos nos quais o publicador também é um assinante (consulte `OtherSubsApenas`)...

Se essa propriedade for omitida, o padrão é o nome de `ReplyToQMgr` no descritor da mensagem (MQMD). Se o nome resultante estiver em branco, ele será padronizado com o nome do gerenciador de filas.

#### **QName (MQPSC\_Q\_NAME)**

O valor é uma sequência contendo o nome da fila de assinantes do publicador, em aplicativos nos quais o publicador também é um assinante (consulte `OtherSubs`).

Se essa propriedade for omitida, o padrão é o nome de `ReplyToQ` no descritor da mensagem (MQMD), o qual não pode estar em branco se `OtherSubsOnly` estiver definido.

## **exemplo**

Alguns exemplos de *NameValueData* para uma mensagem de comando **Publish**.

O primeiro exemplo é para uma publicação enviada pelo simulador de jogo no aplicativo de amostra para indicar que um jogo foi iniciado.

```
<psc>
  <Command>Publish</Command>
  <Topic>Sport/Soccer/Event/MatchStarted</Topic>
</psc>
```

O segundo exemplo é para uma publicação retida. O último placar no jogo entre Team1 e Team2 é publicado.

```
<psc>
  <Command>Publish</Command>
  <PubOpt>RetainPub</PubOpt>
  <Topic>Sport/Soccer/State/LatestScore/Team1 Team2</Topic>
</psc>
```

## **Mensagem Registrar Assinante**

A mensagem de comando **Register Subscriber** é enviada para um gerenciador de filas por um assinante, ou por outro aplicativo em nome de um assinante, para indicar que ele deseja assinar um ou mais tópicos em um ponto de assinatura. Um filtro de conteúdo de mensagem também pode ser especificado.

Em expressões de filtro de publicação / assinatura, aninhar parênteses faz com que o desempenho diminua exponencialmente. Evite aninhar parênteses para uma profundidade maior que cerca de 6.

A mensagem é enviada para o `SYSTEM.BROKER.CONTROL.QUEUE`, que é a fila de controle do gerenciador de filas. A autoridade para colocar uma mensagem nessa fila é necessária, além de autoridade de acesso (configurada pelo administrador do sistema do gerenciador de filas) para o tópico ou tópicos na assinatura.

Se o usuário tiver autoridade em alguns, mas não em todos, os tópicos, somente aqueles com autoridade serão registrados; uma resposta de aviso indica aqueles que não estão registrados.

Consulte [“Configurações de MQMD em mensagens de comando para o gerenciador de filas”](#) na página 929 para obter detalhes dos parâmetros do descritor de mensagens (MQMD) que são necessários ao enviar uma mensagem de comando para o gerenciador de filas.

Se a fila de resposta for uma fila dinâmica temporária, a assinatura será cancelada automaticamente pelo gerenciador de fila quando a fila for fechada.

## Propriedades

### Comando (*MQPSC\_COMMAND*)

O valor é RegSub (*MQPSC\_REGISTER\_ASSINANTE*). Essa propriedade precisa ser especificada.

### Tópico (*MQPSC\_TOPIC*)

O tópico para o qual o assinante deseja receber publicações.. Caracteres curinga podem ser especificados como parte do tópico.

Se você usar o comando do MQSC **display sub** para examinar a assinatura criada dessa maneira, o valor da tag < Topic> será mostrado como a propriedade TOPICSTR da assinatura.

Essa propriedade é necessária e, opcionalmente, pode ser repetida para quantos tópicos forem necessários

### SubPoint (*MQPSC\_SUBSCRIPTION\_POINT*).

O valor é o ponto de assinatura ao qual a assinatura está anexada.

Se essa propriedade for omitida, o ponto de assinatura padrão será usado

No WebSphere Event Broker 6.0, o valor da propriedade <SubPoint> deve corresponder ao valor do atributo do Ponto de Assinatura dos nós de Publicação que estão inscritos.

No IBM WebSphere MQ 7.0.1, o valor da propriedade <SubPoint> deve corresponder ao nome de um ponto de assinatura. Consulte [Incluindo um ponto de assinatura..](#)

### Filtro (*MQPSC\_FILTER*)

O valor é uma expressão SQL que é usada como um filtro sobre o conteúdo de mensagens de publicação. Se uma publicação no tópico especificado corresponder ao filtro, ela será enviada ao assinante. Essa propriedade corresponde à Sequência de Seleção usada nas chamadas MQSUB e MQOPEN. Para obter mais informações, consulte [Selecionando o conteúdo de uma mensagem ..](#)

Se essa propriedade for omitida, nenhuma filtragem de conteúdo ocorrerá..

### RegOpt (*MQPSC\_REGISTRATION\_OPTION*)

Esta propriedade Opções de Registro pode usar os seguintes valores:

#### AddName

(*MQPSC\_ADD\_NAME*)

Quando especificado para uma assinatura existente que corresponde à identidade tradicional desse comando Register Subscription, mas sem nenhum valor atual SubName , o SubName especificado nesse comando é incluído na assinatura.

Se AddName for especificado, o campo SubName será obrigatório, caso contrário, MQRCCF\_REG\_OPTIONS\_ERROR será retornado.

#### CorrelAsId

(*MQPSC\_CORREL\_ID\_AS\_IDENTITY*)

O CorrelId no descritor de mensagem (MQMD) é usado ao enviar publicações correspondentes para a fila de assinantes. O CorrelId não deve ser zero,

#### FullResp

(*MQPSC\_FULL\_RESPONSE*)

Quando especificado, todos os atributos da assinatura serão retornados na mensagem de resposta, se o comando não falhar

FullResp é válido apenas quando a mensagem do comando refere-se a uma única assinatura. Portanto, apenas um tópico é permitido no comando; caso contrário, o comando falhará com o código de retorno *MQRCCF\_REG\_OPTIONS\_ERROR*.

### **InformIfReet**

(*MQPSC\_INFORM\_IF\_\_NAME*)

O gerenciador de filas informa ao assinante se uma publicação é retida quando ele envia uma mensagem de Publicação em resposta a uma mensagem de comando **Register Subscriber** ou **Request Update**. O gerenciador de filas faz isso incluindo a opção de publicação *IsRetainedPub* na mensagem.

### **JoinExcl**

(*MQPSC\_JOIN\_EXCLUSIVE*)

Esta opção indica que o *SubIdentity* especificado deve ser incluído como o membro exclusivo do conjunto de identidades para a assinatura e que nenhuma outra identidade pode ser incluída no conjunto.

Se a identidade já tiver sido associada 'compartilhada' e for a única entrada no conjunto, o conjunto será alterado para um bloqueio exclusivo mantido por essa identidade. Caso contrário, se a subscrição tiver atualmente outras identidades no conjunto de identidade (com acesso compartilhado), o comando falhará com o código de retorno *MQRCCF\_SUBSCRIPTION\_IN\_USE*.

### **JoinShared**

(*MQPSC\_JOIN\_SHARED*)

Esta opção indica que o *SubIdentity* especificado deve ser incluído no conjunto de identidade para a assinatura.

Se a assinatura estiver atualmente bloqueada exclusivamente (usando a opção *JoinExcl*), o comando falhará com o código de retorno *MQRCCF\_SUBSCRIPTION\_LOCKED*, a menos que a identidade que possui a assinatura bloqueada seja a mesma identidade que aquela nessa mensagem de comando. Nesse caso, o bloqueio é modificado automaticamente para um bloqueio compartilhado.

### **Local**

(*MQPSC\_LOCAL*)

A assinatura é local e não é distribuída para outros gerenciadores de fila na rede. As publicações feitas em outros gerenciadores de fila não são entregues a esse assinante, a menos que ele também tenha uma assinatura global correspondente.

### **NewPubsApenas**

(*MQPSC\_NEW\_PUBS\_ONLY*)

As publicações retidas que existem no momento em que a assinatura é registrada não são enviadas para o assinante; apenas novas publicações são enviadas.

Se um assinante registrar novamente e alterar essa opção para que ela não seja mais configurada, uma publicação que já tenha sido enviada para ele poderá ser enviada novamente.

### **NoAlter**

(*MQPSC\_NO\_ALTER*)

Os atributos de uma assinatura correspondente existente não são mudados.

Quando uma assinatura está sendo criada, esta opção é ignorada. Todas as outras opções especificadas aplicam-se à nova assinatura.

Se uma *SubIdentity* também tiver uma das opções de junção (*JoinExcl* ou *JoinShared*) especificado, a identidade será incluída no conjunto de identidade, independentemente de *NoAlter* ser especificado.

### **Nenhum**

(*MQPSC\_NONE*)

Todas as opções de registro assumem seus valores padrão

Se o assinante já estiver registrado, suas opções serão reconfiguradas para seus valores padrão (observe que isso não tem o mesmo efeito que a omissão da propriedade de opções de registro) e a expiração da assinatura será atualizada a partir do MQMD da mensagem **Register Subscriber**.

Se outras opções de registro forem especificadas simultaneamente, Nenhum será ignorado.

#### **NonPers**

(MQPSC\_NON\_PERSISTENT)

As publicações que correspondem a esta assinatura são entregues ao assinante como mensagens não persistentes.

#### **Pers**

(MQPSC\_PERSISTENT)

As publicações que correspondem a essa assinatura são entregues ao assinante como mensagens persistentes

#### **PersAsPub**

(MQPSC\_PERSISTENT\_AS\_PUBLISH)

As publicações que correspondem a esta assinatura são entregues ao assinante com a persistência especificada pelo publicador. Esse não é o comportamento padrão.

#### **PersAsPersAs**

(MQPSC\_PERSISTENT\_AS\_Q)

As publicações correspondentes a esta assinatura são entregues para o assinante com a persistência especificada na fila de assinantes

#### **PubOnReqOnly**

(MQPSC\_PUB\_ON\_REQUEST\_ONLY)

O gerenciador de filas não envia publicações para o assinante, exceto em resposta a uma mensagem de comando **Request Update**

#### **ID doVariableUser**

(MQPSC\_VARIABLE\_USER\_ID)

Quando especificado, a identidade do assinante (fila, gerenciador de filas e correlid) não é restrita a um único ID do usuário... Isso difere do comportamento existente do gerenciador de filas que associa o ID do usuário da mensagem de registro original à identidade do assinante e, a partir daí, evita que qualquer outro usuário use essa identidade. Se um novo assinante tentar usar a mesma identidade, *MQRCCF\_DUPLICATE\_SUBSCRIPTION* será retornado

Isso permite que qualquer usuário modifique ou remova o registro da assinatura se o usuário tiver autoridade adequada.. Portanto, não é necessário verificar se o ID do usuário corresponde ao do assinante original.

Para incluir essa opção em uma assinatura existente, o comando deve vir do mesmo ID do usuário da própria assinatura original.

Se a assinatura do comando **Request Update** tiver `VariableUserId` configurado, isso deverá ser configurado no momento da atualização da solicitação para indicar a qual assinatura é referida. Caso contrário, o ID do usuário do comando **Request Update** será usado para identificar a subscrição. Isso será substituído, juntamente com os outros identificadores de assinante, se um nome de assinatura for fornecido

Se uma mensagem de comando **Register Subscriber** sem esse conjunto de opções se referir a uma assinatura existente que possui essa opção configurada, a opção será removida dessa assinatura e o ID do usuário da assinatura agora será corrigido. Se já existir um assinante que tenha a mesma identidade (fila, gerenciador de fila e identificador de correlação), mas com um ID do usuário diferente associado a ele, o comando falhará com o código de retorno

*MQRCCF\_DUPLICATE\_IDENTITY* porque pode haver apenas um ID do usuário associado a uma identidade do assinante.

Se a propriedade de opções de registro for omitida e o assinante já estiver registrado, suas opções de registro não serão mudadas e a expiração da assinatura será atualizada do MQMD da mensagem **Register Subscriber**.

Se o assinante ainda não estiver registrado, uma nova assinatura será criada com todas as opções de registro usando seus valores padrão.

Os valores padrão são `PersAsPub` e nenhuma outra opção configurada.

#### **QMgrName (MQPSC\_Q\_MGR\_NAME)**

O valor é o nome do gerenciador de filas para a fila de assinantes, para a qual publicações correspondentes são enviadas pelo gerenciador de filas.

Se essa propriedade for omitida, o padrão é o nome de `ReplyToQMgr` no descritor da mensagem (MQMD). Se o nome resultante estiver em branco, ele será padronizado como `QMgrNamedo` gerenciador de filas.

#### **QName (MQPSC\_Q\_NAME)**

O valor é o nome da fila de assinantes para a qual as publicações correspondentes são enviadas pelo gerenciador de filas.

Se essa propriedade for omitida, o padrão será o nome `ReplyToQ` no descritor de mensagens (MQMD), que não deve ficar em branco neste caso...

Se a fila for uma fila dinâmica temporária, entrega não persistente de publicações ( `NonPers` ) deve ser especificado na propriedade `< RegOpt >`

Se a fila for uma fila dinâmica temporária, a assinatura será cancelada automaticamente pelo gerenciador de filas quando a fila for fechada.

#### **SubName (MQPSC\_SUBSCRIPTION\_NAME)**

Este é um nome dado a uma assinatura específica. É possível usá-lo em vez do gerenciador de filas, da fila e do `CorrelId` opcional para referir-se a uma assinatura

Se uma assinatura já existir com esse **SubName**, quaisquer outros atributos da assinatura (Tópico, `QMgrName`, `QName`, `CorrelId`, `UserId` `RegOpts`, `UserSube` `Expiração`) serão substituídos pelos atributos, se especificados, que são transmitidos na nova mensagem de comando `Registrar assinante`. No entanto, se o **SubName** for usado sem nenhum campo `QName` especificado e um `ReplyToQ` for especificado no cabeçalho MQMD, a fila de assinantes será alterada para ser `ReplyToQ`.

Se uma assinatura correspondente à identidade tradicional desse comando já existir, mas não tiver **SubName**, o comando `Registration` falhará com o código de retorno *MQRCCF\_DUPLICATE\_SUBSCRIPTION*, a menos que a opção **AddName** seja especificada.

Se você tentar alterar uma assinatura nomeada existente usando outro comando `Registrar assinante` que especifica o mesmo **SubName** e os valores de `Topic`, `QMgrName`, `QName` e `CorrelId` no novo comando correspondem a uma assinatura existente diferente, com ou sem um `SubName` definido, o comando falhará com o código de retorno *MQRCCF\_DUPLICATE\_SUBSCRIPTION*. Isso evita dois nomes de assinatura que se referem à mesma assinatura.

#### **SubIdentity (MQPSC\_SUBSCRIPTION\_IDENTITY)**

Essa sequência é usada para representar um aplicativo com um interesse em uma assinatura. Ela é uma sequência de caracteres de comprimento variável com um comprimento máximo de 64 caracteres e é opcional. O gerenciador de filas mantém um conjunto de identidades de assinante para cada subscrição. Cada assinatura pode permitir que seu conjunto de identidade contenha apenas uma identidade ou um número ilimitado de identidades (consulte as opções **JoinShared** e **JoinExcl**).

Um comando `subscribe` que especifica a opção **JoinShared** ou **JoinExcl** inclui a **SubIdentity** no conjunto de identidade da assinatura, se ela ainda não estiver lá e se o conjunto de identidades existente permitir tal ação; ou seja, nenhum outro assinante se associou exclusivamente ou o conjunto de identidade está vazio.

Qualquer alteração dos atributos da assinatura como resultado de um comando `Register Subscription` no qual um **SubIdentity** é especificado, somente será bem-sucedido se ele for o único membro do conjunto de identidades para essa assinatura. Caso contrário, o comando falhará com o código de retorno `MQRCCF_SUBSCRIPTION_IN_USE`. Isso impede que os atributos de uma assinatura sejam alterados sem que outros assinantes interessados estejam cientes.

Se você especificar uma sequência de caracteres maior que 64 caracteres, o comando falhará com o código de retorno `MQRCCF_SUB_IDENTITY_ERROR`.

#### **SubUserDados (MQPSC\_SUBSCRIPTION\_USER\_DATA)**

Esta é uma cadeia de texto de comprimento variável. O valor é armazenado pelo gerenciador de filas com a assinatura, mas não tem influência na entrega de publicação para o assinante. O valor pode ser alterado registrando novamente na mesma assinatura com um novo valor. Esse atributo está lá para o uso do aplicativo.

Os **SubUserSubusuário** são retornados nas informações do Metatópico (`MQCACF_REG_SUB_USER_DATA`) para uma assinatura, se presente..

Se você especificar mais de um dos valores da opção de registro `NonPers`, `PersAsPub`, `PersAsQueue`, and `Pers`, então apenas o último será usado. Não é possível combinar essas opções em uma assinatura individual..

### **exemplo**

Aqui está um exemplo de `NameValueData` para uma mensagem de comando **Register Subscriber**.. No aplicativo de amostra, o serviço de resultados usa essa mensagem para registrar uma assinatura para os tópicos que contêm as pontuações mais recentes em todas as correspondências, com a opção 'Persistente como publicação' configurada. A identidade do assinante, incluindo o `CorrelId`, é obtida dos padrões no MQMD.

```
<psc>
  <Command>RegSub</Command>
  <RegOpt>PersAsPub</RegOpt>
  <RegOpt>CorrelAsId</RegOpt>
  <Topic>Sport/Soccer/State/LatestScore/#</Topic>
</psc>
```

### **Mensagem de atualização de solicitação**

A mensagem de comando **Request Update** é enviada de um assinante para um gerenciador de filas, para solicitar as publicações retidas atuais para o tópico e o ponto de assinatura especificados que correspondem ao filtro especificado (opcional).

Esta mensagem é enviada para o `SYSTEM.BROKER.CONTROL.QUEUE`, a fila de controle do gerenciador de fila. A autoridade para colocar uma mensagem nesta fila é necessária, além da autoridade de acesso para o tópico na atualização da solicitação; isso é configurado pelo administrador do sistema do gerenciador de filas.

Esse comando é normalmente usado se o assinante especificou a opção `PubOnReqOnly` quando registrado. Se o gerenciador de filas tiver quaisquer publicações retidas correspondentes, elas serão enviadas para o assinante. Se o gerenciador de filas não tiver publicações retidas correspondentes, a solicitação falhará com o código de retorno `MQRCCF_NO_RETAINED_MSG`.. O solicitante deve ter registrado anteriormente uma assinatura com os mesmos valores de Tópico, SubPonto Filtro..

### **Propriedades**

#### **Comando (MQPSC\_COMMAND)**

O valor é `ReqUpdate` (`MQPSC_REQUEST_UPDATE`). Essa propriedade precisa ser especificada.

#### **Tópico (MQPSC\_TOPIC)**

O valor é o tópico que o assinante está solicitando; caracteres curinga são permitidos.

Esta propriedade deve ser especificada, mas apenas uma ocorrência é permitida nesta mensagem.

**SubPoint (MQPSC\_SUBSCRIPTION\_POINT).**

O valor é o ponto de assinatura ao qual a assinatura está anexada.

Se essa propriedade for omitida, o ponto de assinatura padrão será usado

**Filtro (MQPSC\_FILTER)**

O valor é uma expressão ESQL usada como um filtro no conteúdo de mensagens de publicação. Se uma publicação no tópico especificado corresponder ao filtro, ela será enviada ao assinante.

A propriedade < Filter> deve ter o mesmo valor especificado na assinatura original para a qual você está agora solicitando uma atualização.

Se essa propriedade for omitida, nenhuma filtragem de conteúdo ocorrerá..

**RegOpt (MQPSC\_REGISTRATION\_OPTION)**

A propriedade de opções de registro pode ter o seguinte valor:

**CorrelAsId**

(MQPSC\_CORREL\_ID\_AS\_IDENTITY)

O CorrelId no descritor de mensagens (MQMD), que não deve ser zero, é usado ao enviar publicações correspondentes à fila de assinantes.

**Nenhum**

(MQPSC\_NONE)

Todas as opções assumem seus valores padrão. Isso tem o mesmo efeito que a omissão da propriedade <RegOpt> . Se outras opções forem especificadas ao mesmo tempo, Nenhuma será ignorada.

**ID doVariableUser**

(MQPSC\_VARIABLE\_USER\_ID)

Quando especificado, a identidade do assinante (fila, gerenciador de filas e correlid) não é restrita a um único ID do usuário.. Isso difere do comportamento existente do gerenciador de filas que associa o ID do usuário da mensagem de registro original à identidade do assinante e, a partir daí, evita que qualquer outro usuário use essa identidade. Se um novo assinante tentar usar a mesma identidade, o comando falhará com o código de retorno *MQRCCF\_DUPLICATE\_SUBSCRIPTION*.

Isso permite que qualquer usuário modifique ou remova o registro da assinatura quando tiver autoridade adequada. Portanto, não há necessidade de verificar se o ID do usuário corresponde ao do assinante original

Para incluir essa opção em uma assinatura existente, o comando deve vir do mesmo ID do usuário da assinatura original.

Se a assinatura do comando **Request Update** tiver *VariableUserId* configurado, isso deverá ser configurado no momento da atualização da solicitação para indicar a qual assinatura é referida. Caso contrário, o ID do usuário do comando **Request Update** será usado para identificar a subscrição Isso será substituído, juntamente com os outros identificadores de assinante, se um nome de assinatura for fornecido

O padrão, se essa propriedade for omitida, é que nenhuma opção de registro está configurada

**QMgrName (MQPSC\_Q\_MGR\_NAME)**

O valor é o nome do gerenciador de filas para a fila de assinantes, para o qual a publicação retida correspondente é enviada pelo gerenciador de filas..

Se essa propriedade for omitida, o padrão é o nome de *ReplyToQMgr* no descritor da mensagem (MQMD). Se o nome resultante estiver em branco, ele será padronizado como *QMgrNamedo* gerenciador de filas.

**QName (MQPSC\_Q\_NAME)**

O valor é o nome da fila de assinantes para a qual a publicação retida correspondente é enviada pelo gerenciador de filas.

Se essa propriedade for omitida, o padrão será o nome ReplyToQ no descritor de mensagens (MQMD), que não deve ficar em branco neste caso...

### **SubName (MQPSC\_SUBSCRIPTION\_NAME)**

Este é um nome dado a uma assinatura específica. Se especificado em um comando **Request Update**, o valor SubName terá precedência sobre todos os outros campos de identificador, exceto o ID do usuário, a menos que VariableUserId seja configurado na própria assinatura. Se VariableUserId não estiver configurado, o comando *Request Update* será bem-sucedido apenas se o ID do usuário da mensagem de comando corresponder ao da assinatura. Se o ID do usuário da mensagem de comando não corresponder ao da assinatura, o comando falhará com o código de retorno *MQRCCF\_DUPLICATE\_IDENTITY*.

Se o VariableUserId estiver configurado e o ID do usuário for diferente daquele da assinatura, o comando será bem-sucedido se o ID do usuário da nova mensagem de comando tiver autoridade para procurar a fila de fluxo e colocar na fila de assinantes da assinatura. Caso contrário, o comando falhará com o código de retorno *MQRCCF\_NOT\_AUTHORIZED*.

Se existir uma assinatura correspondente à identidade tradicional desse comando, mas não tiver SubName, o comando **Request Update** falhará com o código de retorno *MQRCCF\_SUB\_NAME\_ERROR*.

Se for feita uma tentativa de solicitar uma atualização para uma assinatura que tenha um SubName usando uma mensagem de comando que corresponda à identidade tradicional, mas sem SubName especificado, o comando será bem-sucedido.

### **exemplo**

Aqui está um exemplo de NameValueData para uma mensagem de comando **Request Update**. No aplicativo de amostra, o serviço de resultados usa essa mensagem para solicitar publicações retidas contendo as últimas pontuações para todas as equipes. A identidade do assinante, incluindo o CorrelId, é obtida dos padrões no MQMD.

```
<psc>
  <Command>ReqUpdate</Command>
  <RegOpt>CorrelAsId</RegOpt>
  <Topic>Sport/Soccer/State/LatestScore/#</Topic>
</psc>
```

## **Mensagem de Resposta do Gerenciador de Filas**

Uma mensagem **Queue Manager Response** é enviada de um gerenciador de filas para o ReplyToQ de um publicador ou assinante, para indicar o sucesso ou a falha de uma mensagem de comando recebida pelo gerenciador de filas, se o descritor de mensagem de comando especificou que uma resposta é necessária.

A mensagem de resposta está contida no campo NameValueData do cabeçalho MQRFH2, em uma pasta `<pscr>`.

No caso de um aviso ou erro, a mensagem de resposta contém a pasta `<psc>` da mensagem de comando, bem como a pasta `<pscr>`. Os dados da mensagem, se houver, não estão contidos na mensagem de resposta do gerenciador de filas. No caso de um erro, nada da mensagem que causou o erro foi processado; no caso de um aviso, parte da mensagem pode ter sido processada com êxito.

Se houver uma falha ao enviar uma resposta:

- Para mensagens de publicação, o gerenciador de filas tentará enviar a resposta para a fila de mensagens não entregues do IBM MQ se o MQPUT falhar. Isso permite que a publicação seja enviada aos assinantes mesmo que a resposta não possa ser enviada de volta ao publicador.
- Para outras mensagens, ou se a resposta da publicação não conseguir ser enviada à fila dead-letter, um erro será registrado e a mensagem de comando será retrocedida normalmente. Se isso acontece depende de como o nó MQInput tiver sido configurado.



## Propriedades

### Conclusão (*MQPSCR\_CONCLUSÃO*)

O código de conclusão, que pode ter um dos três valores:

#### ok

Comando concluído com êxito

#### aviso

Comando concluído mas com aviso

#### Erro

Comando falhou

### Resposta (*MQPSCR\_RESPONSE*)

A resposta a uma mensagem de comando, se esse comando produziu um código de conclusão de `warning` ou `error`. Ele contém uma propriedade `< Reason>` e pode conter outras propriedades que indicam a causa do aviso ou do erro

No caso de um ou mais erros, somente há uma pasta `response`, indicando somente a causa do primeiro erro. No caso de um ou mais avisos, há uma pasta `response` para cada aviso.

### Razão (*MQPSCR\_REASON*)

O código de razão qualificando o código de conclusão, se este for um `warning` ou `error`. Ele é configurado para um dos códigos de erro listados no exemplo a seguir: A propriedade `< Motivo>` está contida em uma pasta `< Responder>`. O código de razão pode ser seguido por qualquer propriedade válida da pasta `<psc>` (por exemplo, um nome de tópico), indicando a causa do erro ou aviso. Se você obtiver um código de razão de `???`, verifique os dados para correção, por exemplo, sinais de maior e menor correspondentes (`<>`).

## Exemplos

A seguir estão alguns exemplos de `NameValueData` em uma mensagem **Queue Manager Response**. Uma resposta com êxito poderia ser assim:

```
<pscr>
  <Completion>ok</Completion>
</pscr>
```

Eis um exemplo de uma resposta com falha; a falha é um erro de filtro. A primeira sequência `NameValueData` contém a resposta; a segunda contém o comando original.

```
<pscr>
  <Completion>error</Completion>
  <Response>
    <Reason>3150</Reason>
  </Reponse>
</pscr>

<psc>
  ...
  command message (to which
  the queue manager is responding)
  ...
</psc>
```

Eis um exemplo de uma resposta de aviso (devido a tópicos não autorizados). A primeira sequência `NameValueData` contém a resposta; a segunda sequência `NameValueData` contém o comando original.

```
<pscr>
  <Completion>warning</Completion>
  <Response>
    <Reason>3081</Reason>
    <Topic>topic1</Topic>
  </Reponse>
  <Response>
    <Reason>3081</Reason>
```

```

<Topic>topic2</Topic>
</Reponse>
</pscr>

<psc>
...
command message (to which
the queue manager is responding)
...
</psc>

```

## Códigos de Razão de Publicação / Assinatura

Esses códigos de razão podem ser retornados na pasta Motivo de uma resposta de publicação / assinatura <pscr> . Constantes que podem ser usadas para representar esses códigos nas linguagens de programação C ou C++ também são listadas.

As constantes MQRC\_ requerem o arquivo de cabeçalho IBM MQ cmqc.h . As constantes MQRCCF\_ requerem o arquivo de cabeçalho IBM MQ cmqcf.h (além de MQRCCF\_FILTER\_ERROR e MQRCCF\_ERRADA, que requerem o arquivo de cabeçalho cmqpsc.h ).

Código de razão e texto.	Explicação	Emitido por
2336 MQRC_RFH_COMMAND_ERROR	Os valores válidos para o campo < Command> de uma pasta <psc> são: RegSub, DeregSub, Publish, DeletePub, ReqUpdate. Quaisquer outros valores resultarão na emissão desse código de erros	Qualquer comando
2337 MQRC_RFH_PARM_ERROR	As pastas <psc> e <mcd> possuem um conjunto de parâmetros válidos que podem ser especificados nelas. Verifique as descrições dessas pastas e certifique-se de não ter especificado parâmetros incorretos.	Qualquer comando
2338 MQRC_RFH_DUPLICATE_PARM	Alguns parâmetros (por exemplo, Tópico) em uma pasta <psc> podem ser repetidos, mas outros (por exemplo, Comando) não podem ser repetidos. Verifique se você não duplicou um parâmetro não repetido	Qualquer comando
2339 MQRC_RFH_PARM_MISSING	Alguns parâmetros dentro das pastas <psc> ou <mcd> são opcionais e podem ser omitidos; alguns são obrigatórios e não devem ser omitidos. Verifique se você incluiu todos os parâmetros obrigatórios nas pastas <psc> e <mcd> .	Qualquer comando
2551 MQRC_SELECTION_NOT_AVAILABLE	Nenhum provedor de seleção de mensagem estendida estava disponível para determinar quais assinantes com um filtro especificado devem receber a publicação	Publicar, Registrar o Assinante e Solicitar Atualização
	Nenhum provedor de seleção de mensagem estendida estava disponível para manipular o filtro do assinante especificado	Registrar Assinante e Solicitar Atualização

<b>Código de razão e texto.</b>	<b>Explicação</b>	<b>Emitido por</b>
2554 MQRCCF_CONTENT_ERROR	Um provedor de seleção de mensagem estendida localizou um erro na publicação atual ou retida	Atualização de Publicação e Solicitação
3008 MQRCCF_COMMAND_FAILED	Ocorreu um erro interno que impediu a execução correta do comando. O erro poderá ocorrer se o comando for emitido novamente O log de eventos do sistema do gerenciador de filas contém informações que devem ser usadas ao relatar o problema para IBM.	Qualquer comando
3072 MQRCCF_TOPIC_ERROR	Um ou mais dos valores fornecidos para o parâmetro Topic estão incorretos. Verifique se seus valores para o Tópico estão em conformidade com as restrições especificadas	Qualquer comando
3073 MQRCCF_NOT_REGISTERED	A combinação de SubPoint, Tópico e Filtro que você especificou em seu comando DeregSub ou ReqUpdate não era uma combinação com a qual você havia se registrado anteriormente ou, para o comando DeregSub se a opção DeregAll foi especificada, uma das propriedades SubPoint, Tópico ou Filtro não foi usada para remover o registro de nenhuma assinatura.	Comandos Remover Registro do Assinante e Solicitar Atualização
3074 MQRCCF_Q_MGR_NAME_ERROR	O gerenciador de filas especificado não era válido, ou o gerenciador de fila não estava disponível ou não existia	Comandos de Remover Registro do Assinante, Publicar, Registrar Assinante e Solicitar Atualização
3076 MQRCCF_Q_NAME_ERROR	O nome da fila especificado não era válido ou a fila não existia no gerenciador de fila especificado..	Comandos de Remover Registro do Assinante, Publicar, Registrar Assinante e Solicitar Atualização
3077 MQRCCF_NO_RETAINED_MSG	Não houve mensagens retidas para o tópico especificado. Isso pode ou não ser um erro, dependendo do design de seu programa de aplicativo	Comando Solicitar Atualização

<b>Código de razão e texto.</b>	<b>Explicação</b>	<b>Emitido por</b>
3079 MQRCCF_INCORRECT_Q	Os comandos RegSub, DeregSub e ReqUpdate são sempre enviados ao SYSTEM.BROKER.CONTROL.QUEUE do gerenciador de filas para o qual eles são destinados. Comandos de Publicação e Exclusão de Publicação são enviados para a fila de entrada para o fluxo de mensagens de publicação / assinatura específico para o qual eles são destinados; isso é determinado quando o fluxo de mensagens é projetado. Esse código de erro será retornado se um comando for enviado para a fila errada.	Qualquer comando
3080 MQRCCF_CORREL_ID_ERROR	Você especificou CorrelAsId como um de seus parâmetros RegOpt. No entanto, o campo CorrelId do MQMD não contém um identificador de correlação válido (ou seja, é configurado como MQCI_NONE).	Comandos Remover Registro de Assinante e Registrar Assinante
3081 MQRCCF_NOT_AUTHORIZED	Você não está autorizado a executar a ação solicitada. As configurações de autorização para o gerenciador de fila são manipuladas pelo administrador do sistema usando o editor Hierarquia de tópicos.	Comandos de Publicação e Registro de Assinante
3083 MQRCCF_REG_OPTIONS_ERROR	Você especificou um parâmetro RegOpt não reconhecido na pasta <psc> contendo seu comando RegSub ou DeregSub.	Comandos Remover Registro de Assinante e Registrar Assinante
3084 MQRCCF_PUB_OPTIONS_ERROR	Você especificou um parâmetro PubOpt não reconhecido na pasta <psc> que contém seu comando Publish..	comando de publicação
3087 MQRCCF_DEL_OPTIONS_ERROR	Você especificou um parâmetro DelOpt não reconhecido na pasta <psc> que contém seu comando DeletePub.	comando Excluir Publicação
3150 MQRCCF_FILTER_ERROR	O valor especificado para o parâmetro Filter não é válido. Verifique a seção que descreve a sintaxe válida para expressões de filtro e assegure-se de que sua expressão esteja em conformidade.	Comandos Remover Registro do Assinante, Registrar Assinante e Solicitar Atualização
3151 MQRCCF_EQUIVOC_USER	Uma assinatura que corresponde à especificada já existe; no entanto, ela foi registrada por um usuário diferente. Uma assinatura só pode ser alterada ou cancelada pelo usuário que a registrou originalmente.	Comandos Remover Registro do Assinante, Registrar Assinante e Solicitar Atualização

Código de razão e texto.	Explicação	Emitido por
3152 MQRCCF_DUPLICATE_SUBSCRIPTION	Já existe uma assinatura correspondente com um nome de assinatura diferente.	
3153 MQRCCF_SUB_NAME_ERROR	O formato do nome da assinatura não é válido ou uma assinatura correspondente já existe sem nome de assinatura.	
3154 MQRCCF_SUB_IDENTITY_ERROR	O parâmetro de identidade da assinatura está incorreto. O valor fornecido excede o comprimento máximo permitido ou a identidade da assinatura não é atualmente membro do conjunto de identidade da assinatura e uma opção de registro de Junção não foi especificada.	
3155 MQRCCF_SUBSCRIPTION_IN_USE	Uma tentativa de modificar ou remover o registro de uma assinatura foi tentada por um membro do conjunto de identidade quando ele não era o único membro desse conjunto	
3156 MQRCCF_SUBSCRIPTION_LOCKED	A assinatura está bloqueada com exclusividade no momento por outra identidade.	
3157 MQRCCF_ALREADY_ENTROU	Uma opção Join registration foi especificada mas a identidade do assinante já era um membro do conjunto de identidades da assinatura.	

## Configurações de MQMD em mensagens de comando para o gerenciador de filas

Aplicativos que enviam mensagens de comando para o gerenciador de filas usam as seguintes configurações de campos no descritor de mensagens (MQMD). Os campos que são deixados como o valor padrão ou que podem ser configurados para qualquer valor válido da maneira usual não são listados aqui.

### Report

Consulte `MsgType` e `CorrelId`.

### MsgType

`MsgType` deve ser configurado como `MQMT_REQUEST` ou `MQMT_DATAGRAM`.

`MQRC_MSG_TYPE_ERROR` será retornado se `MsgType` não estiver configurado como um desses valores.

`MsgType` deve ser configurado como `MQMT_REQUEST` para uma mensagem de comando se uma resposta for sempre necessária. As sinalizações `MQRO_PAN` e `MQRO_NAN` no campo Relatório não são significativas neste caso.

Se `MsgType` for configurado como `MQMT_DATAGRAM`, as respostas dependerão da configuração dos sinalizadores `MQRO_PAN` e `MQRO_NAN` no campo Relatório :

- `MQRO_PAN` sozinho significa que o gerenciador de filas envia uma resposta apenas se o comando for bem-sucedido

- MQRO\_NAN sozinho significa que o Gerenciador de Filas envia uma resposta apenas se o comando falhar
- Se um comando for concluído com um aviso, uma resposta será enviada se MQRO\_PAN ou MQRO\_NAN estiver configurado..
- MQRO\_PAN + MQRO\_NAN significa que o gerenciador de filas envia uma resposta se o comando for bem-sucedido ou falhar.. Isso tem o mesmo efeito da perspectiva do gerenciador de filas que configurar MsgType para MQMT\_REQUEST.
- Se nem MQRO\_PAN nem MQRO\_NAN forem configurados, nenhuma resposta será enviada.

**Format**

Configure como MQFMT\_RF\_HEADER\_2

**MsgId**

Esse campo é normalmente configurado como MQMI\_NONE, para que o gerenciador de filas gere um valor exclusivo.

**CorrelId**

Esse campo pode ser configurado para qualquer valor. Se a identidade do emissor incluir um CorrelId, especifique esse valor, juntamente com MQRO\_PASS\_CORREL\_ID no campo Relatório, para assegurar que ele seja configurado em todas as mensagens de resposta enviadas pelo gerenciador de filas ao emissor.

**ReplyToQ**

Este campo define a fila para a qual respostas, se houver, devem ser enviadas. Essa pode ser a fila do emissor; isso tem a vantagem de o parâmetro QName poder ser omitido da mensagem. Se, no entanto, as respostas forem enviadas para uma fila diferente, o parâmetro QName será necessário.

**ReplyToQMgr**

Este campo define o gerenciador de filas para respostas. Se você deixar esse campo em branco (o valor padrão), o gerenciador de fila local colocará seu próprio nome nesse campo...

**Configurações do MQMD para publicações encaminhadas por um gerenciador de filas**

Um gerenciador de filas utiliza essas configurações de campos no descritor de mensagens (MQMD) quando ele envia uma publicação para um assinante. Todos os outros campos no MQMD são configurados para os seus valores padrão.

**Report**

O Relatório é configurado como MQRO\_NONE

**MsgType**

MsgType é configurado como MQMT\_DATAGRAM.

**Expiração**

Expiração é configurado para o valor na mensagem Publicar recebida do publicador. No caso de uma mensagem retida, o tempo pendente é reduzido pelo tempo aproximado em que a mensagem esteve no gerenciador de filas.

**Format**

Formato é configurado como MQFMT\_RF\_HEADER\_2

**MsgId**

MsgId é configurado para um valor exclusivo.

**CorrelId**

Se CorrelId fizer parte da identidade do assinante, este será o valor especificado pelo assinante ao registrar. Caso contrário, será um valor diferente de zero escolhido pelo gerenciador de filas.

**prioridade**

Prioridade usa o valor configurado pelo publicador ou conforme resolvido se o publicador especificou MQPRI\_PRIORITY\_AS\_Q\_DEF.

**Persistência**

Persistência usa o valor configurado pelo publicador ou conforme resolvido se o publicador especificou MQPER\_PERSISTENCE\_AS\_Q\_DEF, a menos que especificado de outra forma na mensagem Registrar assinante para o assinante para o qual esta publicação está sendo enviada.

**ReplyToQ**

ReplyToQ é configurado como espaços em branco

**ReplyToQMgr**

ReplyToQMgr é configurado para o nome do gerente de filas.

**UserIdentifier**

UserIdentifier é o identificador do usuário do assinante, conforme configurado quando o assinante é registrado.

**AccountingToken**

AccountingToken é o token de contabilidade do assinante, conforme configurado quando o assinante foi registrado pela primeira vez.

**AppIdentityData**

AppIdentity são os dados de identidade do aplicativo do assinante, conforme configurado quando o assinante foi registrado pela primeira vez

**PutAppType**

PutAppType é configurado como MQAT\_BROKER.

**PutAppName**

PutAppName é configurado para os primeiros 28 caracteres do nome do gerenciador de filas..

**PutDate**

PutDate é a data em que a mensagem foi colocada

**PutTime**

PutTime é o horário em que a mensagem foi colocada

**AppOriginData**

AppOrigin são configurados como espaços em branco

**Configurações do MQMD em mensagens de resposta do gerenciador de filas**

Um gerenciador de filas utiliza essas configurações de campos no descritor de mensagens (MQMD) ao enviar uma resposta para uma mensagem de publicação.. Todos os outros campos no MQMD são configurados para os seus valores padrão

**Report**

Relatório é configurado para todos os zeros..

**MsgType**

MsgType é configurado como MQMT\_REPLY.

**Format**

Formato é configurado como MQFMT\_RF\_HEADER\_2

**MsgId**

A configuração de MsgId depende das opções de Relatório na mensagem de comando original Por padrão, ele é configurado como MQMI\_NONE, para que o gerenciador de filas gere um valor exclusivo.

**CorrelId**

A configuração de CorrelId depende das opções de Relatório na mensagem de comando original Por padrão, isso significa que o CorrelId é configurado para o mesmo valor que o MsgId da mensagem de comando. Isso pode ser usado para correlacionar comandos com as suas respostas

**prioridade**

Prioridade é configurado para o mesmo valor da mensagem de comando original.

**Persistência**

Persistência é configurado com o valor configurado na mensagem de comando original.

### **Expiração**

Expiração é configurado para o mesmo valor que na mensagem de comando original recebida pelo gerenciador de fila..

### **PutAppType**

PutAppType é configurado como MQAT\_BROKER.

### **PutAppName**

PutAppName é configurado para os primeiros 28 caracteres de nome do gerenciador de filas.

Outros campos de contexto são configurados como se gerados com MQPMO\_PASS\_IDENTITY\_CONTEXT.

## **Codificações da máquina**

Esta seção descreve a estrutura do campo *Encoding* no descritor de mensagens..

Consulte “MQMD - descritor de mensagem” na página 435 para um resumo dos campos na estrutura.

O campo *Encoding* é um número inteiro de 32 bits que é dividido em quatro subcampos separados; esses subcampos identificam:

- A codificação usada para números inteiros binários
- A codificação usada para números inteiros decimais compactados
- A codificação usada para números de vírgula flutuante
- Bits reservados

Cada subcampo é identificado por uma máscara de bits que tem 1 bits nas posições correspondentes ao subcampo e 0 bits em outro lugar. Os bits são numerados de forma que o bit 0 é o bit mais significativo e o bit 31 o bit menos significativo. As máscaras a seguir são definidas:

### **MQENC\_INTEGER\_MASK**

Máscara para codificação de número inteiro binário.

Este subcampo ocupa posições de bits 28 a 31 no campo *Encoding* .

### **MQENC\_DECIMAL\_MASK**

Máscara para codificação de número inteiro decimal compactado.

Esse subcampo ocupa as posições de bits 24 a 27 no campo *Encoding* .

### **MQENC\_FLOAT\_MASK**

Máscara para a codificação de vírgula flutuante

Este subcampo ocupa posições de bits de 20 a 23 no campo *Encoding* .

### **MQENC\_RESERVED\_MASK**

Máscara para bits reservados..

Esse subcampo ocupa as posições de bits de 0 a 19 no campo *Encoding*

## **Codificação de número inteiro binário**

Os valores a seguir são válidos para a codificação de número inteiro binário:

### **MQENC\_INTEGER\_UNDEFINED**

Os números inteiros binários são representados usando uma codificação indefinida.

### **MQENC\_INTEGER\_NORMAL**

Os números inteiros binários são representados da maneira convencional:

- O byte menos significativo no número tem o endereço mais alto de qualquer um dos bytes no número; o byte mais significativo tem o endereço mais baixo
- O bit menos significativo em cada byte é adjacente ao byte com o próximo endereço superior; o bit mais significativo em cada byte é adjacente ao byte com o próximo endereço inferior



### **MQENC\_INTEGER\_REVERSED**

Os números inteiros binários são representados da mesma maneira que MQENC\_INTEGER\_NORMAL, mas com os bytes organizados em ordem inversa... Os bits em cada byte são organizados da mesma maneira que MQENC\_INTEGER\_NORMAL.

## **Codificação de número inteiro decimal compactado**

Os valores a seguir são válidos para a codificação de número inteiro decimal compactada:

### **MQENC\_DECIMAL\_UNDEFINED**

Números inteiros decimais compactados são representados usando uma codificação indefinida.

### **MQENC\_DECIMAL\_NORMAL**

Números inteiros decimais compactados são representados da maneira convencional:

- Cada dígito decimal na forma de impressão do número é representado em decimal compactado por um único dígito hexadecimal no intervalo de X' 0 'a X' 9'. Cada dígito hexadecimal ocupa quatro bits e, portanto, cada byte no número decimal compactado representa dois dígitos decimais no formato imprimível do número
- O byte menos significativo no número decimal compactado é o byte que contém o dígito decimal menos significativo. Dentro desse byte, os quatro bits mais significativos contêm o dígito decimal menos significativo e os quatro bits menos significativos contêm o sinal. O sinal é X'C '(positivo), X'D' (negativo) ou X'F' (não assinado).
- O byte menos significativo no número tem o endereço mais alto de qualquer um dos bytes no número; o byte mais significativo tem o endereço mais baixo.
- O bit menos significativo em cada byte é adjacente ao byte com o próximo endereço superior; o bit mais significativo em cada byte é adjacente ao byte com o próximo endereço inferior.

### **MQENC\_DECIMAL\_REVERSED**

Números inteiros decimais compactados são representados da mesma maneira que MQENC\_DECIMAL\_NORMAL, mas com os bytes organizados em ordem inversa. Os bits dentro de cada byte são organizados da mesma maneira que MQENC\_DECIMAL\_NORMAL

## **Codificação de ponto flutuante**

Os valores a seguir são válidos para a codificação de vírgula flutuante:

### **MQENC\_FLOAT\_UNDEFINED**

Os números de ponto flutuante são representados usando uma codificação indefinida.

### **MQENC\_FLOAT\_IEEE\_NORMAL**

Os números de vírgula flutuante são representados usando o IEEE padrão<sup>4</sup>Formato de vírgula flutuante, com os bytes organizados da seguinte forma:

- O byte menos significativo na mantissa tem o endereço mais alto de qualquer um dos bytes no número; o byte que contém o expoente tem o endereço mais baixo
- O bit menos significativo em cada byte é adjacente ao byte com o próximo endereço superior; o bit mais significativo em cada byte é adjacente ao byte com o próximo endereço inferior

Detalhes da codificação flutuante IEEE podem ser encontrados na norma IEEE 754.

### **MQENC\_FLOAT\_IEEE\_REVERSED**

Os números de vírgula flutuante são representados da mesma maneira que MQENC\_FLOAT\_IEEE\_NORMAL, mas com os bytes organizados em ordem inversa Os bits dentro de cada byte são organizados da mesma maneira que MQENC\_FLOAT\_IEEE\_NORMAL

### **MQENC\_FLOAT\_S390**

Os números de vírgula flutuante são representados usando o formato de vírgula flutuante padrão System/390 ; também é usado por System/370.

---

<sup>4</sup> O Instituto de Engenheiros Elétricos e Eletrônicos

## Construindo codificações

Para construir um valor para o campo *Encoding* no MQMD, as constantes relevantes que descrevem as codificações necessárias podem ser incluídas juntas (não inclua a mesma constante mais de uma vez) ou combinadas usando a operação OR bit a bit (se a linguagem de programação suportar operações de bit).

Qualquer que seja o método utilizado, combine apenas uma das codificações MQENC\_INTEGER\_\* com uma das codificações MQENC\_DECIMAL\_\*, e uma das codificações MQENC\_FLOAT\_\*.

## Analisando codificações

O campo *Encoding* contém subcampos; por isso, aplicativos que precisam examinar a codificação de número inteiro, decimal compactado ou flutuante devem usar uma das técnicas descritas.

## Usando operações de bit

Se a linguagem de programação suportar operações de bits, execute as seguintes etapas:

1. Selecione um dos seguintes valores, de acordo com o tipo de codificação necessário:
  - MQENC\_INTEGER\_MASK para a codificação de número inteiro binário
  - MQENC\_DECIMAL\_MASK para a codificação de número inteiro decimal compactado
  - MQENC\_FLOAT\_MASK para a codificação de ponto flutuante,Chame o valor A.
2. Combine o campo *Encoding* com A usando a operação AND bitwise; chame o resultado B.
3. B é a codificação necessária e pode ser testada para igualdade com cada um dos valores válidos para esse tipo de codificação.

## Usando aritmética

Se a linguagem de programação *não* suportar operações de bits, execute as etapas a seguir usando aritmética de número inteiro:

1. Selecione um dos seguintes valores, de acordo com o tipo de codificação necessário:
  - 1 para a codificação de número inteiro binário
  - 16 para a codificação de número inteiro decimal compactado
  - 256 para a codificação de ponto flutuanteChame o valor A.
2. Divida o valor do campo *Encoding* por A ; Chame o resultado B.
3. Divida B por 16; chame o resultado de C
4. Multiplique C por 16 e subtraia de B ; Chame o resultado D.
5. Multiplique D por A ; e Chame o resultado E.
6. E é a codificação necessária e pode ser testada para igualdade com cada um dos valores válidos para esse tipo de codificação.

## Resumo de codificações de arquitetura de máquina..

As codificações para arquiteturas de máquina são mostradas em [Tabela 631 na página 935](#)

Tabela 631. Resumo de codificações para arquiteturas de máquina..

Arquitetura da máquina	Codificação de número inteiro binário	Codificação de número inteiro decimal compactada	Codificação de Ponto Flutuante
IBM i	normal	normal	IEEE normal
Intel x86	reversed	reversed	IEEE revertido
PowerPC	normal	normal	IEEE normal
System/390	normal	normal	System/390

## Opções de relatório e sinalizações de mensagem

Esta seção descreve os campos *Report* e *MsgFlags* que fazem parte do MQMD do descritor de mensagem especificado nas chamadas MQGET, MQPUT e MQPUT1

Os tópicos nesta seção descrevem:

- A estrutura do campo de relatório e como o gerenciador de filas o processa
- Como um aplicativo analisa o campo de relatório
- A estrutura do campo message-flags

Para obter mais informações sobre o descritor de mensagens MQMD, consulte [“MQMD - descritor de mensagem”](#) na página 435..

### Estrutura do campo de relatório

Essas informações descrevem a estrutura do campo de relatório

O campo *Report* é um número inteiro de 32 bits que é dividido em três subcampos separados. Estes subcampos identificam:

- Opções de relatório que serão rejeitadas se o gerenciador de fila local não as reconhecer
- Opções de relatório que são sempre aceitas, mesmo que o gerenciador de fila local não as reconheça
- Opções de relatório que são aceitas somente se determinadas outras condições forem satisfeitas

Cada subcampo é identificado por uma máscara de bits que tem 1 bits nas posições correspondentes ao subcampo e 0 bits em outro lugar. Os bits em um subcampo não são necessariamente adjacentes. Os bits são numerados de forma que o bit 0 é o bit mais significativo e o bit 31 o bit menos significativo. As máscaras a seguir são definidas para identificar os subcampos:

#### MQRO\_REJECT\_UNSUP\_MASK

Essa máscara identifica as posições de bits dentro do campo *Report* em que as opções de relatório que não são suportadas pelo gerenciador de filas locais fazem com que a chamada MQPUT ou MQPUT1 falhe com o código de conclusão MQCC\_FAILED e o código de razão MQRC\_REPORT\_OPTIONS\_ERROR.

Este subcampo ocupa posições de bits 3 e 11 a 13.

#### MQRO\_ACCEPT\_UNSUP\_MASK

Essa máscara identifica as posições de bit dentro do campo *Report* em que as opções de relatório que não são suportadas pelo gerenciador de filas locais são, no entanto, aceitas nas chamadas MQPUT ou MQPUT1 ... Código de conclusão MQCC\_WARNING com código de razão MQRC\_UNKNOWN\_REPORT\_OPTION são retornados neste caso.

Este subcampo ocupa posições de bits 0 a 2, 4 a 10 e 24 a 31.

As seguintes opções de relatório estão incluídas neste subcampo:

- MQRO\_ACTIVITY
- MQRO\_COPY\_MSG\_ID\_TO\_CORREL\_ID

- MQRO\_DEAD\_LETTER\_Q
- MQRO\_DISCARD\_MSG
- MQRO\_EXCEPTION
- MQRO\_EXCEPTION\_WITH\_DATA
- MQRO\_EXCEPTION\_WITH\_FULL\_DATA
- MQRO\_EXPIRATION
- MQRO\_EXPIRATION\_WITH\_DATA
- MQRO\_EXPIRATION\_WITH\_FULL\_DATA
- MQRO\_NAN
- MQRO\_NEW\_MSG\_ID
- MQRO\_NONE
- MQRO\_PAN
- MQRO\_PASS\_CORREL\_ID
- MQRO\_PASS\_MSG\_ID

### **MQRO\_ACCEPT\_UNSUP\_IF\_XMIT\_MASK**

Essa máscara identifica as posições de bit dentro do campo *Report* em que as opções de relatório que não são suportadas pelo gerenciador de filas locais são, no entanto, aceitas nas chamadas MQPUT ou MQPUT1 *fornecidas* que ambas as condições a seguir são satisfeitas:

- A mensagem destina-se a um gerenciador de filas remotas
- O aplicativo não está colocando a mensagem diretamente em uma fila de transmissão local (ou seja, a fila identificada pelos campos *ObjectQMgrName* e *ObjectName* no descritor de objeto especificado na chamada MQOPEN ou MQPUT1 não é uma fila de transmissão local).

O código de conclusão MQCC\_WARNING com o código de razão MQRC\_UNKNOWN\_REPORT\_OPTION será retornado se essas condições forem satisfeitas e MQCC\_FAILED com o código de razão MQRC\_REPORT\_OPTIONS\_ERROR se não.

Este subcampo ocupa posições de bits 14 a 23.

As seguintes opções de relatório estão incluídas neste subcampo:

- MQRO\_COA
- MQRO\_COA\_WITH\_DATA
- MQRO\_COA\_WITH\_FULL\_DATA
- MQRO\_COD
- MQRO\_COD\_WITH\_DATA
- MQRO\_COD\_WITH\_FULL\_DATA

Se quaisquer opções forem especificadas no campo *Report* que o gerenciador de filas não reconhece, o gerenciador de filas verifica cada subcampo por vez usando a operação AND bit a bit para combinar o campo *Report* com a máscara para esse subcampo. Se o resultado dessa operação não for zero, o código de conclusão e os códigos de razão descritos anteriormente serão retornados.

Se MQCC\_WARNING for retornado, não será definido qual código de razão será retornado se existirem outras condições de aviso..

A capacidade de especificar e ter opções de relatório aceitas que não são reconhecidas pelo gerenciador de filas locais é útil ao enviar uma mensagem com uma opção de relatório reconhecida e processada por um gerenciador de filas *remotas* .

## **Analisando o campo de relatório**

O campo *Report* contém subcampos; devido a isso, os aplicativos que precisam verificar se o emissor da mensagem solicitou um relatório específico devem usar uma das técnicas descritas

## Usando operações de bit

Se a linguagem de programação suportar operações de bits, execute as seguintes etapas:

1. Selecione um dos seguintes valores, de acordo com o tipo de relatório a ser verificado:

- Relatório MQRO\_COA\_WITH\_FULL\_DATA para COA
- Relatório MQRO\_COD\_WITH\_FULL\_DATA para COD
- MQRO\_EXCEPTION\_WITH\_FULL\_DATA para relatório de exceções
- MQRO\_EXPIRATION\_WITH\_FULL\_DATA para relatório de expiração

Chame o valor A.

No z/OS, use os valores MQRO\_\*\_WITH\_DATA em vez dos valores MQRO\_\*\_WITH\_FULL\_DATA.

2. Combine o campo *Report* com A usando a operação AND bitwise; chame o resultado B.

3. Teste B para igualdade com cada valor possível para esse tipo de relatório.

Por exemplo, se A for MQRO\_EXCEPTION\_WITH\_FULL\_DATA, teste B para igualdade com cada um dos seguintes para determinar o que foi especificado pelo emissor da mensagem:

- MQRO\_NONE
- MQRO\_EXCEPTION
- MQRO\_EXCEPTION\_WITH\_DATA
- MQRO\_EXCEPTION\_WITH\_FULL\_DATA

Os testes podem ser executados em qualquer ordem que seja mais conveniente para a lógica do aplicativo

Use um método semelhante para testar para as opções MQRO\_PASS\_MSG\_ID ou MQRO\_PASS\_CORREL\_ID; selecione como o valor A o que for apropriado dessas duas constantes e, em seguida, continue conforme descrito anteriormente.

## Usando aritmética

Se a linguagem de programação *não* suportar operações de bits, execute as etapas a seguir usando aritmética de número inteiro:

1. Selecione um dos seguintes valores, de acordo com o tipo de relatório a ser verificado:

- Relatório MQRO\_COA para COA
- Relatório MQRO\_COD para COD
- MQRO\_EXCEPTION para relatório de exceções
- MQRO\_EXPIRATION para relatório de expiração

Chame o valor A.

2. Divida o campo *Report* por A ; Chame o resultado B.

3. Dividir B por 8 ; Chame o resultado C.

4. Multiplicar C por 8 e subtrair de B ; Chame o resultado D.

5. Multiplique D por A ; e Chame o resultado E.

6. Teste E para igualdade com cada valor possível para esse tipo de relatório.

Por exemplo, se A for MQRO\_EXCEPTION, teste E para igualdade com cada um dos itens a seguir para determinar o que foi especificado pelo emissor da mensagem:

- MQRO\_NONE
- MQRO\_EXCEPTION
- MQRO\_EXCEPTION\_WITH\_DATA
- MQRO\_EXCEPTION\_WITH\_FULL\_DATA

Os testes podem ser executados em qualquer ordem que seja mais conveniente para a lógica do aplicativo

O pseudocódigo a seguir ilustra essa técnica para mensagens de relatório de exceção:

```
A = MQRO_EXCEPTION
B = Report/A
C = B/8
D = B - C*8
E = D*A
```

Use um método semelhante para testar para as opções MQRO\_PASS\_MSG\_ID ou MQRO\_PASS\_CORREL\_ID; selecione como o valor A o que for apropriado e, em seguida, continue conforme descrito anteriormente, mas substituindo o valor 8 nas etapas anteriores pelo valor 2.

## Estrutura do campo message-flags

Estas informações descrevem a estrutura do campo message-flags.

O campo *MsgFlags* é um número inteiro de 32 bits que é dividido em três subcampos separados. Estes subcampos identificam:

- Sinalizadores de mensagem que serão rejeitadas se o gerenciador de filas locais não os reconhecer
- Sinalizadores de mensagens que são sempre aceitos, mesmo se o gerenciador da fila local não os reconhecer
- Sinalizadores de mensagem que são aceitos apenas se determinadas outras condições forem satisfeitas..

**Nota:** Todos os subcampos em *MsgFlags* são reservados para uso pelo gerenciador de fila..

Cada subcampo é identificado por uma máscara de bits que tem 1 bits nas posições correspondentes ao subcampo e 0 bits em outro lugar. Os bits são numerados de forma que o bit 0 é o bit mais significativo e o bit 31 o bit menos significativo. As máscaras a seguir são definidas para identificar os subcampos:

### MQMF\_REJECT\_UNSUP\_MASK

Essa máscara identifica as posições de bits dentro do campo *MsgFlags* em que os sinalizadores de mensagens que não são suportados pelo gerenciador de filas locais causam a falha da chamada MQPUT ou MQPUT1 com o código de conclusão MQCC\_FAILED e o código de razão MQRC\_MSG\_FLAGS\_ERROR.

Este subcampo ocupa posições de bits 20 a 31.

Os sinalizadores de mensagem a seguir estão incluídos neste subcampo:

- MQMF\_LAST\_MSG\_IN\_GROUP
- MQMF\_LAST\_SEGMENT
- MQMF\_MSG\_IN\_GROUP
- MQMF\_SEGMENT
- MQMF\_SEGMENTATION\_ALLOWED
- MQMF\_SEGMENTATION\_INIBIDA

### MQMF\_ACCEPT\_UNSUP\_MASK

Essa máscara identifica as posições de bits dentro do campo *MsgFlags* em que os sinalizadores de mensagem que não são suportados pelo gerenciador de filas locais são, no entanto, aceitos nas chamadas MQPUT ou MQPUT1 O código de conclusão é MQCC\_OK..

Este subcampo ocupa posições de bits de 0 a 11.

### MQMF\_ACCEPT\_UNSUP\_IF\_XMIT\_MASK

Essa máscara identifica as posições de bits no campo *MsgFlags* em que sinalizadores de mensagens que não são suportados pelo gerenciador de filas locais são, no entanto, aceitos nas chamadas MQPUT ou MQPUT1 *forneidas* que ambas as condições a seguir são satisfeitas:

- A mensagem destina-se a um gerenciador de filas remotas
- O aplicativo não está colocando a mensagem diretamente em uma fila de transmissão local (ou seja, a fila identificada pelos campos *ObjectQMgrName* e *ObjectName* no descritor de objeto especificado na chamada MQOPEN ou MQPUT1 não é uma fila de transmissão local.

O código de conclusão MQCC\_OK será retornado se essas condições forem satisfeitas e MQCC\_FAILED com o código de razão MQRC\_MSG\_FLAGS\_ERROR se não.

Este subcampo ocupa posições de bits 12 a 19.

Se houver sinalizadores especificados no campo *MsgFlags* que o gerenciador de filas não reconhece, o gerenciador de filas verifica cada subcampo por vez usando a operação AND bit a bit para combinar o campo *MsgFlags* com a máscara para esse subcampo. Se o resultado dessa operação não for zero, o código de conclusão e os códigos de razão descritos anteriormente serão retornados.

## saída de conversão de dados

Esta coleção de tópicos descreve a interface para a saída de conversão de dados e o processamento executado por o gerenciador de filas quando a conversão de dados é necessária

Para obter mais informações sobre a conversão de dados, consulte *Conversão de dados em IBM MQ* em <https://www.ibm.com/support/pages/node/317869>

A saída de conversão de dados é chamada como parte do processamento da chamada MQGET para converter os dados da mensagem do aplicativo na representação requerida pelo aplicativo de recebimento. A conversão dos dados da mensagem do aplicativo é opcional; ela requer que a opção MQGMO\_CONVERT seja especificada na chamada MQGET

Os seguintes assuntos são descritos:

- O processamento executado pelo Gerenciador de Filas em resposta à opção MQGMO\_CONVERT; consulte [“Processamento de conversão” na página 939](#)
- Convenções de processamento usadas pelo gerenciador de fila ao processar um formato integrado; essas convenções são recomendadas para saídas gravadas pelo usuário também Consulte o [“Convenções de processamento” na página 941](#).
- Considerações especiais para converter mensagens de relatório; consulte [“Conversão de mensagens de relatório” na página 945](#).
- Os parâmetros transmitidos para a saída de conversão de dados; consulte [“MQ\\_DATA\\_CONV\\_EXIT-Saída de conversão de dados” na página 958](#)
- Uma chamada que pode ser usada a partir da saída para converter dados de caracteres entre diferentes representações. Consulte [“MQXCNV-Converter caracteres” na página 952](#)
- O parâmetro data-structure que é específico da saída; consulte [“MQDXP-Parâmetro de saída de conversão de dados” na página 946](#).

## Processamento de conversão

Essas informações descrevem o processamento executado pelo gerenciador de filas em resposta à opção MQGMO\_CONVERT..

O gerenciador de filas executa as ações a seguir se a opção MQGMO\_CONVERT for especificada na chamada MQGET e houver uma mensagem a ser retornada para o aplicativo:

1. Se um ou mais dos seguintes itens for verdadeiro, nenhuma conversão será necessária:
  - Os dados da mensagem já estão no conjunto de caracteres e na codificação necessários pelo aplicativo que emite a chamada MQGET. O aplicativo deve configurar os campos *CodedCharSetId* e *Encoding* no parâmetro **MsgDesc** da chamada MQGET para os valores necessários antes de emitir a chamada.
  - O comprimento dos dados da mensagem é zero.
  - O comprimento do parâmetro **Buffer** da chamada MQGET é zero.

Nesses casos, a mensagem é retornada sem conversão para o aplicativo emitindo a chamada MQGET; os valores *CodedCharSetId* e *Encoding* no parâmetro **MsgDesc** são configurados para os valores nas informações de controle na mensagem e a chamada é concluída com uma das seguintes combinações de código de conclusão e código de razão:

Tabela 632. Combinações de código de conclusão e de código de razão

Código de conclusão	Código de razão
MQCC_OK	MQRC_NONE
MQCC_WARNING	MQRC_TRUNCATED_MSG_ACCEPTED
MQCC_WARNING	MQRC_TRUNCATED_MSG_FAILED

As etapas a seguir serão executadas somente se o conjunto de caracteres ou a codificação de dados da mensagem for diferente do valor correspondente no parâmetro **MsgDesc** e houver dados a serem convertidos:

2. Se o campo *Format* nas informações de controle na mensagem tiver o valor MQFMT\_NONE, a mensagem será retornada não convertida, com código de conclusão MQCC\_WARNING e código de razão MQRC\_FORMAT\_ERROR.

Em todos os outros casos, o processamento de conversão continua

3. A mensagem é removida da fila e colocada em um buffer temporário que tem o mesmo tamanho que o parâmetro **Buffer**. Para operações de navegação, a mensagem é copiada no buffer temporário, em vez de ser removida da fila.
4. Se a mensagem tiver que ser truncada para caber no buffer, o seguinte será feito:
  - Se a opção MQGMO\_ACCEPT\_TRUNCATED\_MSG não tiver sido especificada, a mensagem será retornada sem conversão, com código de conclusão MQCC\_WARNING e código de razão MQRC\_TRUNCATED\_MSG\_FAILED.
  - Se a opção MQGMO\_ACCEPT\_TRUNCATED\_MSG *tiver sido* especificada, o código de conclusão será configurado como MQCC\_WARNING, o código de razão será configurado como MQRC\_TRUNCATED\_MSG\_ACCEPTED e o processamento de conversão continuará
5. Se a mensagem puder ser acomodada no buffer sem truncamento ou a opção MQGMO\_ACCEPT\_TRUNCATED\_MSG foi especificada, o seguinte será feito:
  - Se o formato for integrado, o buffer será transmitido para o serviço de conversão de dados do gerenciador de filas.
  - Se o formato não for um formato integrado, o buffer será transmitido para uma saída gravada pelo usuário com o mesmo nome que o formato. Se a saída não puder ser localizada, a mensagem será retornada sem conversão, com código de conclusão MQCC\_WARNING e código de razão MQRC\_FORMAT\_ERROR.

Se nenhum erro ocorrer, a saída do serviço de conversão de dados ou da saída gravada pelo usuário será a mensagem convertida, além do código de conclusão e do código de razão a ser retornado ao aplicativo que emite a chamada MQGET.

6. Se a conversão for bem-sucedida, o gerenciador de fila retorna a mensagem convertida para o aplicativo. Nesse caso, o código de conclusão e o código de razão retornados pela chamada MQGET são uma das seguintes combinações:

Tabela 633. Combinações de código de conclusão e de código de razão

Código de conclusão	Código de razão
MQCC_OK	MQRC_NONE
MQCC_WARNING	MQRC_TRUNCATED_MSG_ACCEPTED

No entanto, se a conversão for executada por uma saída gravada pelo usuário, outros códigos de razão poderão ser retornadas, mesmo quando a conversão for bem-sucedida



Se a conversão falhar, o gerenciador de filas retornará a mensagem não convertida para o aplicativo, com os campos *CodedCharSetId* e *Encoding* no parâmetro **MsgDesc** configurado para os valores nas informações de controle na mensagem, e com o código de conclusão MQCC\_WARNING

## Convenções de processamento

Ao converter um formato integrado, o gerenciador de filas segue as convenções de processamento descritas..

As saídas gravadas pelo usuário também devem seguir essas convenções, embora isso não seja imposto pelo gerenciador de fila Os formatos integrados convertidos pelo gerenciador de filas são:

- MQFMT\_ADMIN
- MQFMT\_CICS ( z/OS apenas)
- MQFMT\_COMMAND\_1
- MQFMT\_COMMAND\_2
- MQFMT\_DEAD\_LETTER\_HEADER
- MQFMT\_DIST\_HEADER
- MQFMT\_EVENT versão 1
- MQFMT\_EVENT versão 2
- MQFMT\_IMS
- MQFMT\_IMS\_VAR\_STRING
- MQFMT\_MD\_EXTENSION
- MQFMT\_PCF
- MQFMT\_REF\_MSG\_HEADER
- MQFMT\_RF\_HEADER
- MQFMT\_RF\_HEADER\_2
- MQFMT\_STRING
- MQFMT\_TRIGGER
- MQFMT\_WORK\_INFO\_HEADER (apenas z/OS )
- MQFMT\_XMIT\_Q\_HEADER

1. Se a mensagem for expandida durante a conversão e exceder o tamanho do parâmetro **Buffer** , o seguinte será feito:

- Se a opção MQGMO\_ACCEPT\_TRUNCATED\_MSG não foi especificada, a mensagem será retornada sem conversão, com o código de conclusão MQCC\_WARNING e o código de razão MQRC\_CONVERTED\_MSG\_TOO\_BIG.
- Se a opção MQGMO\_ACCEPT\_TRUNCATED\_MSG *foi especificada* , a mensagem será truncada, o código de conclusão será configurado como MQCC\_WARNING, o código de razão será configurado como MQRC\_TRUNCATED\_MSG\_ACCEPTED e o processamento de conversão continuará..

2. Se ocorrer truncamento (antes ou durante a conversão), o número de bytes válidos retornados no parâmetro **Buffer** poderá ser menor que o comprimento do buffer.

Isso pode ocorrer, por exemplo, se um número inteiro de 4 bytes ou um caractere DBCS estiver no final do buffer. O elemento incompleto de informações não é convertido, e esses bytes na mensagem retornada não contêm informações válidas Isso também pode ocorrer se uma mensagem que foi truncada antes da conversão for reduzida durante a conversão.

Se o número de bytes válidos retornados for menor que o comprimento do buffer, os bytes não utilizados no final do buffer serão configurados como nulos..

3. Se uma matriz ou sequência estiver no final do buffer, o máximo de dados possível será convertido; apenas o elemento de matriz ou o caractere DBCS específico que está incompleto não será convertido; os elementos ou caracteres da matriz anteriores serão convertidos...

4. Se ocorrer truncamento (antes ou durante a conversão), o comprimento retornado para o parâmetro **DataLength** será o comprimento da mensagem não convertida antes do truncamento.
5. Quando as cadeias são convertidas entre conjuntos de caracteres de byte único (SBCS), conjuntos de caracteres de byte duplo (DBCS) ou conjuntos de caracteres de multibyte (MBCS), as cadeias podem expandir ou contrair.

- Nos formatos PCF MQFMT\_ADMIN, MQFMT\_EVENT, e MQFMT\_PCF, as sequências nas estruturas MQCFST e MQCFSL expandem ou contraem conforme necessário para acomodar a sequência após a conversão.

Para a estrutura MQCFSL da lista de sequências, as sequências na lista podem expandir ou contrair por quantias diferentes. Se isso acontecer, o gerenciador de filas preencherá as sequências mais curtas com espaços em branco para torná-las do mesmo comprimento que a sequência mais longa após a conversão

- No formato MQFMT\_REF\_MSG\_HEADER, as sequências endereçadas pelos campos SrcEnvOffset, SrcNameOffset, DestEnvOffset e DestNameOffset se expandem ou contratam conforme necessário para acomodar as sequências após a conversão
  - No formato MQFMT\_RF\_HEADER, o campo NameValueString expande ou contrai conforme necessário para acomodar os pares nome-valor após a conversão.
  - Em estruturas com tamanhos de campo fixos, o gerenciador de fila permite que sequências sejam expandidas ou contradas dentro de seus campos fixos, desde que nenhuma informação significativa seja perdida. Nesse sentido, os espaços em branco e os caracteres que seguem o primeiro caractere nulo no campo são tratados como insignificantes.
    - Se a sequência for expandida, mas apenas caracteres insignificantes precisarem ser descartados para acomodar a sequência convertida no campo, a conversão será bem-sucedida e a chamada será concluída com MQCC\_OK e o código de razão MQRC\_NONE (assumindo nenhum outro erro).
    - Se a sequência for expandida, mas a sequência convertida requerer que caracteres significativos sejam descartados para se ajustarem no campo, a mensagem será retornada não convertida e a chamada será concluída com MQCC\_WARNING e o código de razão MQRC\_CONVERTED\_STRING\_TOO\_BIG.
- Nota:** O código de razão MQRC\_CONVERTED\_STRING\_TOO\_BIG resulta neste caso se a opção MQGMO\_ACCEPT\_TRUNCATED\_MSG foi especificada.
- Se a sequência se contrai, o gerenciador de filas preenche a sequência com espaços em branco para o comprimento do campo.

6. Para mensagens que consistem em uma ou mais estruturas de cabeçalho do MQ seguidas por dados do usuário, uma ou mais das estruturas de cabeçalho podem ser convertidas, enquanto o restante da mensagem não é. No entanto, (com duas exceções) os campos *CodedCharSetId* e *Encoding* em cada estrutura de cabeçalho sempre indicam corretamente o conjunto de caractere e a codificação dos dados que seguem a estrutura de cabeçalho...

As duas exceções são as estruturas MQCIH e MQIIH, em que os valores nos campos *CodedCharSetId* e *Encoding* nessas estruturas não são significativos. Para essas estruturas, os dados após a estrutura estão no mesmo conjunto de caracteres e na codificação que a própria estrutura MQCIH ou MQIIH.

7. Se os campos *CodedCharSetId* ou *Encoding* nas informações de controle da mensagem que está sendo recuperada ou no parâmetro **MsgDesc**, especificar valores que são indefinidos ou não suportados, o gerenciador de filas poderá ignorar o erro se o valor indefinido ou não suportado não precisar ser usado na conversão da mensagem.

Por exemplo, se o campo *Encoding* na mensagem especificar uma codificação flutuante não suportada, mas a mensagem contiver apenas dados de número inteiro ou contiver dados de vírgula flutuante que não requerem conversão (porque as codificações flutuantes de origem e de destino são idênticas), o erro poderá não ser diagnosticado.

Se o erro for diagnosticado, a mensagem será retornada não convertida, com o código de conclusão MQCC\_WARNING e um dos códigos de razão MQRC\_SOURCE\_\*\_ERROR ou MQRC\_TARGET\_\*

\_ERROR (conforme apropriado); os campos *CodedCharSetId* e *Encoding* no parâmetro **MsgDesc** são configurados para os valores nas informações de controle na mensagem.

Se o erro não for diagnosticado e a conversão for concluída com êxito, os valores retornados nos campos *CodedCharSetId* e *Encoding* no parâmetro **MsgDesc** serão aqueles especificados pelo aplicativo emitindo a chamada MQGET.

8. Em todos os casos, se a mensagem for retornada para o aplicativo não convertido, o código de conclusão será configurado como MQCC\_WARNING e os campos *CodedCharSetId* e *Encoding* no parâmetro **MsgDesc** serão configurados para os valores apropriados para os dados não convertidos. Isso também é feito para MQFMT\_NONE

O parâmetro **Reason** é configurado para um código que indica por que a conversão não pôde ser executada, a menos que a mensagem também tenha que ser truncada; os códigos de razão relacionados ao truncamento têm precedência sobre os códigos de razão relacionados à conversão (Para determinar se uma mensagem truncada foi convertida, verifique os valores retornados nos campos *CodedCharSetId* e *Encoding* no parâmetro **MsgDesc**.)

Quando um erro é diagnosticado, um código de razão específico é retornado ou o código de razão geral MQRC\_NOT\_CONVERTED.. O código de razão retornado depende dos recursos de diagnóstico do serviço de conversão de dados subjacente..

9. Se o código de conclusão MQCC\_WARNING for retornado e mais de um código de razão for relevante, a ordem de precedência será a seguinte:
  - a. As seguintes razões têm precedência sobre todas as outras; apenas uma das razões neste grupo pode surgir:
    - MQRC\_SIGNAL\_REQUEST\_ACCEPTED
    - MQRC\_TRUNCATED\_MSG\_ACCEPTED
  - b. A ordem de precedência dentro dos códigos de razão restantes não está definida

10. Na conclusão da chamada MQGET:

- O código de razão a seguir indica que a mensagem foi convertida com êxito:
  - MQRC\_NONE
- Os seguintes códigos de razão indicam que a mensagem *pode* ter sido convertida com êxito (verifique os campos *CodedCharSetId* e *Encoding* no parâmetro **MsgDesc** para descobrir):
  - MQRC\_MSG\_MARKED\_BROWSE\_CO\_OP
  - MQRC\_TRUNCATED\_MSG\_ACCEPTED
- Todos os outros códigos de razão indicam que a mensagem não foi convertida.

O processamento a seguir é específico para os formatos integrados; ele não se aplica a formatos definidos pelo usuário:

11. Com exceção dos seguintes formatos:

- MQFMT\_ADMIN
- MQFMT\_COMMAND\_1
- MQFMT\_COMMAND\_2
- MQFMT\_EVENT
- MQFMT\_IMS\_VAR\_STRING
- MQFMT\_PCF
- MQFMT\_STRING

nenhum dos formatos integrados pode ser convertido de ou para conjuntos de caracteres que não possuem caracteres SBCS para os caracteres que são válidos em nomes de filas... Se for feita uma tentativa de executar essa conversão, a mensagem será retornada sem conversão, com código de conclusão MQCC\_WARNING e código de razão MQRC\_SOURCE\_CCSID\_ERROR ou MQRC\_TARGET\_CCSID\_ERROR, conforme apropriado.

O conjunto de caracteres Unicode UTF-16 é um exemplo de um conjunto de caracteres que não possui caracteres SBCS para os caracteres que são válidos em nomes de fila.

12. Se os dados da mensagem para um formato integrado forem truncados, os campos na mensagem que contêm comprimentos de cadeias ou contagens de elementos ou estruturas não serão ajustados para refletir o comprimento dos dados realmente retornados para o aplicativo; os valores retornados para esses campos dentro dos dados da mensagem serão os valores aplicáveis à mensagem *antes do truncamento...*

Ao processar mensagens como uma mensagem MQFMT\_ADMIN truncada, assegure-se de que o aplicativo não tente acessar dados além do final dos dados retornados.

13. Se o nome do formato for MQFMT\_DEAD\_LETTER\_HEADER, os dados da mensagem começam com uma estrutura MQDLH, possivelmente seguida por zero ou mais bytes de dados da mensagem do aplicativo. O formato, o conjunto de caracteres e a codificação dos dados da mensagem do aplicativo são definidos pelos campos `Format`, `CodedCharSetId` e `Encoding` na estrutura MQDLH no início de uma mensagem. Como a estrutura MQDLH e os dados da mensagem do aplicativo podem ter conjuntos de caracteres e codificações diferentes, um, outro ou ambos da estrutura MQDLH e os dados da mensagem do aplicativo podem requerer conversão.

O gerenciador de filas converte a estrutura MQDLH primeiro, conforme necessário. Se a conversão for bem-sucedida ou a estrutura MQDLH não precisar de conversão, o gerenciador de filas verificará os campos `CodedCharSetId` e `Encoding` na estrutura MQDLH para ver se a conversão dos dados da mensagem do aplicativo é necessária. Se a conversão for necessária, o gerenciador de filas chamará a saída gravada pelo usuário com o nome fornecido pelo campo `Format` na estrutura MQDLH ou executará a própria conversão (se `Format` for o nome de um formato integrado).

Se a chamada MQGET retornar um código de conclusão de MQCC\_WARNING e o código de razão for um daqueles que indicam que a conversão não foi bem-sucedida, um dos seguintes se aplica:

- Não foi possível converter a estrutura MQDLH. Nesse caso, os dados da mensagem do aplicativo também não terão sido convertidos.
- A estrutura MQDLH foi convertida, mas os dados da mensagem do aplicativo não foram.

O aplicativo pode examinar os valores retornados nos campos `CodedCharSetId` e `Encoding` no parâmetro **MsgDesc** e aqueles na estrutura MQDLH, para determinar qual dos anteriormente se aplica.

14. Se o nome do formato for MQFMT\_XMIT\_Q\_HEADER, os dados da mensagem começam com uma estrutura MQXQH, possivelmente seguida por zero ou mais bytes de dados adicionais. Esses dados adicionais geralmente são os dados da mensagem do aplicativo (que podem ter comprimento zero), mas também pode haver uma ou mais estruturas de cabeçalho MQ adicionais presentes, no início dos dados adicionais.

A estrutura MQXQH deve estar no conjunto de caracteres e codificação do gerenciador de fila. O formato, o conjunto de caracteres e a codificação dos dados após a estrutura MQXQH são fornecidos pelos campos `Format`, `CodedCharSetId` e `Encoding` na estrutura MQMD contida no MQXQH. Para cada estrutura de cabeçalho MQ subsequente presente, os campos `Format`, `CodedCharSetId` e `Encoding` na estrutura descrevem os dados que seguem essa estrutura; esses dados são outra estrutura de cabeçalho do MQ ou os dados da mensagens do aplicativo.

Se a opção MQGMO\_CONVERT for especificada para uma mensagem MQFMT\_XMIT\_Q\_HEADER, os dados da mensagem do aplicativo e algumas das estruturas de cabeçalho do MQ serão convertidas, *mas os dados na estrutura MQXQH não serão*. No retorno da chamada MQGET, portanto:

- Os valores dos campos `Format`, `CodedCharSetId` e `Encoding` no parâmetro **MsgDesc** descrevem os dados na estrutura MQXQH e não os dados da mensagem do aplicativo; portanto, os valores não são os mesmos especificados pelo aplicativo que emitiu a chamada MQGET.

O efeito disso é que um aplicativo que obtém repetidamente mensagens de uma fila de transmissão por meio da opção MQGMO\_CONVERT especificada deve reconfigurar os campos `CodedCharSetId` e `Encoding` no parâmetro **MsgDesc** para os valores necessários para os dados da mensagem do aplicativo, antes de cada chamada MQGET.

- Os valores dos campos `Format`, `CodedCharSetIde` e `Encoding` na última estrutura do cabeçalho do MQ presente descrevem os dados da mensagem do aplicativo. Se não houver outras estruturas de cabeçalho MQ presentes, os dados da mensagem do aplicativo serão descritos por esses campos na estrutura MQMD dentro da estrutura MQXQH. Se a conversão for bem-sucedida, os valores serão os mesmos especificados no parâmetro **MsgDesc** pelo aplicativo que emitiu a chamada MQGET.

Se a mensagem for uma mensagem de lista de distribuição, a estrutura MQXQH será seguida por uma estrutura MQDH (além de suas matrizes de registros MQOR e MQPMR), que por sua vez pode ser seguida por zero ou mais estruturas de cabeçalho MQ adicionais e zero ou mais bytes de dados da mensagem do aplicativo.. Como a estrutura MQXQH, a estrutura MQDH deve estar no conjunto de caracteres e na codificação do gerenciador de filas e não será convertida na chamada MQGET, mesmo que a opção MQGMO\_CONVERT seja especificada.

O processamento das estruturas MQXQH e MQDH descritas anteriormente é destinado principalmente para uso pelos agentes do canal de mensagem quando eles obtêm mensagens de filas de transmissão.

## Conversão de mensagens de relatório

Em geral, uma mensagem de relatório pode conter quantidades variáveis de dados da mensagem do aplicativo, de acordo com as opções de relatório especificadas pelo emissor da mensagem original. No entanto, um relatório de atividades pode conter dados, mas sem a opção de relatório mencionar `*_WITH_DATA` na constante.

Em particular, uma mensagem de relatório pode conter:

1. Nenhum dado de mensagem do aplicativo..

2. Alguns dos dados da mensagem do aplicativo da mensagem original

Isso ocorre quando o emissor da mensagem original especifica `MQRO_*_WITH_DATA` e a mensagem é maior que 100 bytes.

3. Todos os dados da mensagem do aplicativo da mensagem original

Isso ocorre quando o emissor da mensagem original especifica `MQRO*_WITH_FULL_DATA` ou especifica `MQRO*_WITH_DATA` e a mensagem tem 100 bytes ou menos.

Quando o gerenciador de filas ou o agente do canal de mensagens gera uma mensagem de relatório, ele copia o nome do formato da mensagem original no campo `Format` nas informações de controle na mensagem de relatório. O nome do formato na mensagem de relatório pode, portanto, implicar um comprimento de dados diferente do comprimento realmente presente na mensagem de relatório (casos 1 e 2 anteriormente).

Se a opção MQGMO\_CONVERT for especificada quando a mensagem de relatório for recuperada.:

- Para o caso 1 anteriormente, a saída de conversão de dados não é chamada (porque a mensagem de relatório não possui dados)
- Para o caso 3 anteriormente, o nome do formato implica corretamente o comprimento dos dados da mensagem..
- Mas para o caso 2 anteriormente, a saída de conversão de dados é chamada para converter uma mensagem que é *menor* do que o comprimento implícito pelo nome do formato.

Além disso, o código de razão passado para a saída geralmente é `MQRC_NONE` (ou seja, o código de razão não indica que a mensagem foi truncada. Isso acontece porque os dados da mensagem foram truncados pelo *emissor* da mensagem de relatório, e não pelo gerenciador de filas do receptor em resposta à chamada MQGET

Devido a essas possibilidades, a saída de conversão de dados não deve usar o nome do formato para deduzir o comprimento dos dados transmitidos para ela; em vez disso, a saída deve verificar o comprimento de dados fornecidos e estar preparada para converter menos dados do que o comprimento implícito pelo nome do formato. Se os dados puderem ser convertidos com êxito, o código de conclusão

MQCC\_OK e o código de razão MQRC\_NONE devem ser retornados pela saída. O comprimento dos dados da mensagem a serem convertidos é transmitido para a saída como o parâmetro **InBufferLength**.

## Interface de programação sensível ao produto

### MQDXP-Parâmetro de saída de conversão de dados

A estrutura MQDXP é um parâmetro que o gerenciador de filas passa para a saída de conversão de dados quando a saída é chamada para converter os dados da mensagem como parte do processamento da chamada MQGET. Consulte a descrição da chamada MQ\_DATA\_CONV\_EXIT para obter detalhes da saída de conversão de dados..

Os dados de caractere em MQDXP estão no conjunto de caracteres do gerenciador de fila local; isso é fornecido pelo atributo do gerenciador de fila **CodedCharSetId**. Os dados numéricos em MQDXP estão na codificação da máquina nativa; isso é fornecido por MQENC\_NATIVE.

Apenas os campos *DataLength*, *CompCode*, *Reason* e *ExitResponse* em MQDXP podem ser mudados pela saída; as alterações em outros campos são ignoradas. No entanto, o campo *DataLength* não poderá ser alterado se a mensagem que está sendo convertida for um segmento que contém apenas parte de uma mensagem lógica.

Quando o controle retorna para o gerenciador de fila a partir da saída, o gerenciador de filas verifica os valores retornados em MQDXP. Se os valores retornados não forem válidos, o gerenciador de filas continuará processando como se a saída tivesse retornado MQXDR\_CONVERSION\_FAILED em *ExitResponse*; no entanto, o gerenciador de filas ignora os valores dos campos *CompCode* e *Reason* retornados pela saída nesse caso e usa em vez disso os valores que esses campos tinham na *entrada* para a saída. Os valores a seguir em MQDXP fazem com que esse processamento ocorra:

- O campo *ExitResponse* não MQXDR\_OK e não MQXDR\_CONVERSION\_FAILED
- Campo *CompCode* não MQCC\_OK e não MQCC\_WARNING
- *DataLength* campo menor que zero ou *DataLength* campo alterado quando a mensagem que está sendo convertida é um segmento que contém apenas parte de uma mensagem lógica.

A tabela a seguir resume os campos na estrutura

Tabela 634. Campos em MQDXP		
Campo	Descrição	Tópico
<i>StrucId</i>	Identificador de estruturação	<a href="#">StrucId</a>
<i>Version</i>	Número de versão da estrutura	<a href="#">Versão</a>
<i>AppOptions</i>	Opções de aplicativo	<a href="#">AppOptions</a>
<i>Encoding</i>	Codificação numérica requerida pelo aplicativo	<a href="#">Codificação</a>
<i>CodedCharSetId</i>	Conjunto de caracteres requerido pelo aplicativo	<a href="#">CodedCharSetId</a>
<i>DataLength</i>	Comprimento em bytes de dados da mensagem	<a href="#">DataLength</a>
<i>CompCode</i>	Código de conclusão	<a href="#">CompCode</a>
<i>Reason</i>	Código de razão qualificando <i>CompCode</i>	<a href="#">Razão</a>
<i>ExitResponse</i>	Resposta da saída	<a href="#">ExitResponse</a>
<i>Hconn</i>	Identificador de conexões	<a href="#">Hconn</a>
<i>pEntryPoints</i>	Endereço da estrutura MQIEP	<a href="#">pEntryPontos</a>

## Campos

A estrutura MQDXP contém os seguintes campos; os campos são descritos em ordem alfabética.

### AppOptions

Tipo: MQLONG

Esta é uma cópia do campo *Options* da estrutura MQGMO especificada pelo aplicativo que emite a chamada MQGET. A saída pode precisar examiná-los para verificar se a opção MQGMO\_ACCEPT\_TRUNCATED\_MSG foi especificada

Este é um campo de entrada para a saída

### CodedCharSetId

Tipo: MQLONG

Este é o identificador do conjunto de caracteres codificado do conjunto de caracteres requerido pelo aplicativo que emite a chamada MQGET. Consulte o campo *CodedCharSetId* na estrutura MQMD para obter mais detalhes... Se o aplicativo especificar o valor especial MQCCSI\_Q\_MGR na chamada MQGET, o gerenciador de fila mudará isso para o identificador do conjunto de caracteres real do conjunto de caracteres usado pelo gerenciador de filas, antes de chamar a saída...

Se a conversão for bem-sucedida, a saída deverá copiar isso para o campo *CodedCharSetId* no descritor de mensagens.

Este é um campo de entrada para a saída

### CompCode

Tipo: MQLONG

Quando a saída é chamada, ela contém o código de conclusão retornado para o aplicativo que emitiu a chamada MQGET, se a saída não fizer nada. Ele é sempre MQCC\_WARNING, porque a mensagem foi truncada ou a mensagem requer conversão e isso ainda não foi feito

Na saída da saída, esse campo contém o código de conclusão a ser retornado para o aplicativo no parâmetro **CompCode** da chamada MQGET; apenas MQCC\_OK e MQCC\_WARNING são válidos. Consulte a descrição do campo *Reason* para obter as sugestões sobre como a saída pode configurar esse campo na saída

Este é um campo de entrada / saída na saída.

### DataLength

Tipo: MQLONG

Quando a saída for chamada, esse campo conterá o comprimento original dos dados da mensagem do aplicativo. Se a mensagem foi truncada para caber no buffer fornecido pelo aplicativo, o tamanho da mensagem fornecida para a saída será *menor* do que o valor de *DataLength*. O tamanho da mensagem fornecida para a saída é sempre fornecido pelo parâmetro **InBufferLength** da saída, independentemente de qualquer truncamento ocorrido.

O truncamento é indicado pelo campo *Reason* que tem o valor MQRC\_TRUNCATED\_MSG\_ACCEPTED na entrada para a saída

A maioria das conversões não precisa mudar esse comprimento, mas uma saída pode fazer isso, se necessário; o valor configurado pela saída é retornado para o aplicativo no parâmetro **DataLength** da chamada MQGET. No entanto, este comprimento não pode ser alterado se a mensagem que está sendo convertida for um segmento que contém apenas parte de uma mensagem lógica. Isso ocorre porque alterar o comprimento faria com que os deslocamentos de segmentos posteriores na mensagem lógica fossem incorretos.

Observe que, se a saída desejar mudar o comprimento dos dados, esteja ciente de que o gerenciador de fila já decidiu se os dados da mensagem se ajustam no buffer do aplicativo, com base no comprimento dos dados *não convertidos*. Esta decisão determina se a mensagem é removida da fila (ou o cursor de navegação movido, para um pedido de navegação) e não é afetada por qualquer alteração no comprimento de dados causada pela conversão.. Por essa razão, recomenda-se que

as saídas de conversão não causem uma mudança no comprimento dos dados da mensagem do aplicativo

Se a conversão de caracteres implicar uma mudança de comprimento, uma sequência poderá ser convertida em outra sequência com o mesmo comprimento em bytes, truncando espaços em branco à direita ou preenchendo com espaços em branco, conforme necessário.

A saída não será invocada se a mensagem não contiver dados da mensagem do aplicativo; portanto, *DataLength* é sempre maior que zero

Este é um campo de entrada / saída na saída.

### **Encoding**

Tipo: MQLONG

Codificação numérica requerida pelo aplicativo.

Essa é a codificação numérica necessária pelo aplicativo que emite a chamada MQGET; consulte o campo *Encoding* na estrutura MQMD para obter mais detalhes.

Se a conversão for bem-sucedida, a saída copiará isso no campo *Encoding* no descritor de mensagens.

Este é um campo de entrada para a saída

### **ExitOptions**

Tipo: MQLONG

Este é um campo reservado; seu valor é 0.

### **ExitResponse**

Tipo: MQLONG

Resposta da saída. Isto é configurado pela saída para indicar o sucesso ou não da conversão. Ele deve ser um dos seguintes:

#### **MQXDR\_OK**

A conversão foi bem-sucedida.

Se a saída especificar esse valor, o gerenciador de filas retorna o seguinte para o aplicativo que emitiu a chamada MQGET:

- O valor do campo *CompCode* na saída da saída
- O valor do campo *Reason* na saída da saída
- O valor do campo *DataLength* na saída da saída
- O conteúdo do buffer de saída da saída *OutBuffer* O número de bytes retornado é o menor dos parâmetros **OutBufferLength** da saída e o valor do campo *DataLength* na saída da saída.

Se os campos *Encoding* e *CodedCharSetId* no parâmetro do descritor de mensagem da saída estiverem *ambos* inalterados, o gerenciador de filas retornará:

- O valor dos campos *Encoding* e *CodedCharSetId* na estrutura MQDXP na *entrada* para a saída.

Se um ou ambos os campos *Encoding* e *CodedCharSetId* no parâmetro do descritor de mensagem da saída tiverem sido alterados, o gerenciador de filas retornará:

- O valor dos campos *Encoding* e *CodedCharSetId* no parâmetro do descritor de mensagem de saída na saída da saída

#### **MQXDR\_CONVERSION\_FAILED**

A conversão foi malsucedida.

Se a saída especificar esse valor, o gerenciador de filas retorna o seguinte para o aplicativo que emitiu a chamada MQGET:

- O valor do campo *CompCode* na saída da saída
- O valor do campo *Reason* na saída da saída



- O valor do campo *DataLength* na entrada para a saída
- O conteúdo do buffer de entrada da saída *InBuffer*.. O número de bytes retornados é fornecido pelo parâmetro **InBufferLength**

Se a saída alterou *InBuffer*, os resultados serão indefinidos.

*ExitResponse* é um campo de saída da saída.

### Hconn

Tipo: MQHCONN

Este é um identificador de conexão que pode ser usado na chamada MQXCNVC. Esse identificador não é necessariamente igual ao identificador especificado pelo aplicativo que emitiu a chamada MQGET.

### pEntryPoints

Tipo: PMQIEP

O endereço de uma estrutura MQIEP através da qual chamadas MQI e DCI podem ser feitas.

### Reason

Tipo: MQLONG

Código de razão qualificando *CompCode*.

Quando a saída é chamada, isso contém o código de razão que é retornado para o aplicativo que emitiu a chamada MQGET, se a saída optar por não fazer nada. Entre os valores possíveis estão MQRC\_TRUNCATED\_MSG\_ACCEPTED, indicando que a mensagem foi truncada para se ajustar ao buffer fornecido pelo aplicativo e MQRC\_NOT\_CONVERTED, indicando que a mensagem requer conversão, mas que isso ainda não foi feito.

Na saída da saída, esse campo contém a razão para ser retornado ao aplicativo no parâmetro **Reason** da chamada MQGET; o seguinte é recomendado:

- Se *Reason* tivesse o valor MQRC\_TRUNCATED\_MSG\_ACCEPTED na entrada para a saída, os campos *Reason* e *CompCode* não deverão ser alterados, independentemente de a conversão ser bem-sucedida ou falhar.

(Se o campo *CompCode* não for MQCC\_OK, o aplicativo que recupera a mensagem poderá identificar uma falha de conversão comparando os valores *Encoding* e *CodedCharSetId* retornados no descritor de mensagens com os valores solicitados; em contraste, o aplicativo não pode distinguir uma mensagem truncada de uma mensagem que se ajustou ao buffer. Por esse motivo, MQRC\_TRUNCATED\_MSG\_ACCEPTED deve ser retornado em preferência a qualquer uma das razões que indicam falha de conversão.)

- Se *Reason* tivesse qualquer outro valor na entrada para a saída:
  - Se a conversão for bem-sucedida, *CompCode* deverá ser configurado como MQCC\_OK e *Reason* configurado como MQRC\_NONE
  - Se a conversão falhar ou a mensagem expandir e precisar ser truncada para caber no buffer, *CompCode* deverá ser configurado como MQCC\_WARNING (ou deixado inalterado) e *Reason* configurado como um dos valores listados, para indicar a natureza da falha.

Observe que se a mensagem após a conversão for muito grande para o buffer, ela deverá ser truncada apenas se o aplicativo que emitiu a chamada MQGET tiver especificado a opção MQGMO\_ACCEPT\_TRUNCATED\_MSG:

  - Se ela especificou essa opção, a razão MQRC\_TRUNCATED\_MSG\_ACCEPTED será retornada
  - Se não tiver especificado essa opção, a mensagem será retornada sem conversão, com o código de razão MQRC\_CONVERTED\_MSG\_TOO\_BIG.

Os códigos de razão listados são recomendados para uso pela saída para indicar a razão pela qual a conversão falhou, mas a saída pode retornar outros valores do conjunto de códigos MQRC\_\*, se considerado apropriado. Além disso, o intervalo de valores MQRC\_APPL\_FIRST até MQRC\_APPL\_LAST são alocados para uso pela saída para indicar as condições que a saída deseja comunicar com o aplicativo que está emitindo a chamada MQGET.

**Nota:** Se a mensagem não puder ser convertida com êxito, a saída deverá retornar MQXDR\_CONVERSION\_FAILED no campo *ExitResponse*, para fazer com que o gerenciador de filas retorne a mensagem não convertida. Isso é verdadeiro, independentemente do código de razão retornado no campo *Reason*

**MQRC\_APPL\_FIRST**

(900, X'384') Valor mais baixo para o código de razão definido pelo aplicativo.

**MQRC\_APPL\_LAST**

(999, X'3E7') Valor mais alto para o código de razão definido pelo aplicativo.

**MQRC\_CONVERTED\_MSG\_TOO\_BIG**

(2120, X'848') Dados convertidos muito grandes para o buffer.

**MQRC\_NOT\_CONVERTED**

(2119, X'847') Dados da mensagem não convertidos.

**MQRC\_SOURCE\_CCSD\_ERROR**

(2111, X'83F') Identificador do conjunto de caracteres codificados da origem inválido.

**MQRC\_SOURCE\_DECIMAL\_ENC\_ERROR**

(2113, X'841') Codificação decimal compactada na mensagem não reconhecida.

**MQRC\_SOURCE\_FLOAT\_ENC\_ERROR**

(2114, X'842') Codificação de ponto flutuante na mensagem não reconhecida.

**MQRC\_SOURCE\_INTEGER\_ENC\_ERROR**

(2112, X'840') Codificação de número inteiro da origem não reconhecida.

**MQRC\_TARGET\_CCSD\_ERROR**

(2115, X'843') Identificador do conjunto de caracteres codificados do destino inválido.

**MQRC\_TARGET\_DECIMAL\_ENC\_ERROR**

(2117, X'845') Codificação decimal compactada especificada pelo receptor não reconhecida.

**MQRC\_TARGET\_FLOAT\_ENC\_ERROR**

(2118, X'846') Codificação de ponto flutuante especificada pelo receptor não reconhecida.

**MQRC\_TARGET\_INTEGER\_ENC\_ERROR**

(2116, X'844') Codificação de número inteiro do destino não reconhecida.

**MQRC\_TRUNCATED\_MSG\_ACCEPTED**

(2079, X'81F') Mensagem truncada retornada (processamento concluído).

Este é um campo de entrada / saída na saída.

**StrucId**

Tipo: MQCHAR4

Identificador de estruturação.O valor deve ser:.

**MQDXP\_STRUC\_ID**

Identificador da estrutura do parâmetro de saída de conversão de dados.

Para a linguagem de programação C, a constante MQDXP\_STRUC\_ID\_ARRAY também é definida; ela tem o mesmo valor que MQDXP\_STRUC\_ID, mas é uma matriz de caracteres em vez de uma sequência.

Este é um campo de entrada para a saída

**Version**

Tipo: MQLONG

Número de versão da estrutura.O valor deve ser:.

**MQDXP\_VERSION\_1**

Número da versão para a estrutura do parâmetro de saída de conversão de dados

A constante a seguir especifica o número da versão atual:

**MQDXP\_CURRENT\_VERSION**

A versão atual da estrutura do parâmetro de saída de conversão de dados

**Nota:** Quando uma nova versão desta estrutura é introduzida, o layout da peça existente não é alterado. Portanto, a saída deve verificar se o campo *Version* é igual ou maior que a versão mais baixa que contém os campos que a saída precisa usar.

Este é um campo de entrada para a saída

## Declaração C

```
typedef struct tagMQDXP MQDXP;
struct tagMQDXP {
    MQCHAR4  StrucId;           /* Structure identifier */
    MQLONG   Version;          /* Structure version number */
    MQLONG   ExitOptions;      /* Reserved */
    MQLONG   AppOptions;       /* Application options */
    MQLONG   Encoding;         /* Numeric encoding required by
                               application */
    MQLONG   CodedCharSetId;   /* Character set required by application */
    MQLONG   DataLength;       /* Length in bytes of message data */
    MQLONG   CompCode;         /* Completion code */
    MQLONG   Reason;           /* Reason code qualifying CompCode */
    MQLONG   ExitResponse;     /* Response from exit */
    MQHCONN  Hconn;           /* Connection handle */
    PMQIEP   pEntryPoints;     /* Address of the MQIEP structure */
};
```

## Declaração COBOL (somente IBM i)

```
** MQDXP structure
10 MQDXP.
** Structure identifier
15 MQDXP-STRUCID PIC X(4).
** Structure version number
15 MQDXP-VERSION PIC S9(9) BINARY.
** Reserved
15 MQDXP-EXITOPTIONS PIC S9(9) BINARY.
** Application options
15 MQDXP-APPOPTIONS PIC S9(9) BINARY.
** Numeric encoding required by application
15 MQDXP-ENCODING PIC S9(9) BINARY.
** Character set required by application
15 MQDXP-CODEDCHARSETID PIC S9(9) BINARY.
** Length in bytes of message data
15 MQDXP-DATALength PIC S9(9) BINARY.
** Completion code
15 MQDXP-COMPCODE PIC S9(9) BINARY.
** Reason code qualifying COMPCODE
15 MQDXP-REASON PIC S9(9) BINARY.
** Response from exit
15 MQDXP-EXITRESPONSE PIC S9(9) BINARY.
** Connection handle
15 MQDXP-HCONN PIC S9(9) BINARY.
```

## Declaração do assembler System/390

```
MQDXP          DSECT
MQDXP_STRUCID  DS CL4 Structure identifier
MQDXP_VERSION  DS F   Structure version number
MQDXP_EXITOPTIONS DS F   Reserved
MQDXP_APPOPTIONS DS F   Application options
MQDXP_ENCODING DS F   Numeric encoding required by application
MQDXP_CODEDCHARSETID DS F   Character set required by application
MQDXP_DATALength DS F   Length in bytes of message data
MQDXP_COMPCODE DS F   Completion code
MQDXP_REASON   DS F   Reason code qualifying COMPCODE
MQDXP_EXITRESPONSE DS F   Response from exit
MQDXP_HCONN     DS F   Connection handle
*
MQDXP_LENGTH    EQU *-MQDXP
                ORG MQDXP
MQDXP_AREA      DS CL(MQDXP_LENGTH)
```

## MQXCNCV-Converter caracteres

A chamada MQXCNCV converte caracteres de um conjunto de caracteres para outro usando a linguagem de programação C.

Essa chamada faz parte da IBM MQ Data Conversion Interface (DCI), que é uma das interfaces de estrutura IBM MQ .

Nota: A chamada pode ser usada a partir dos ambientes de saída de aplicativo e de conversão de dados.

### Sintaxe

MQXCNCV (*Hconn*, *Opções*, *SourceCCSID*, *SourceLength*, *SourceBuffer*, *TargetCCSID*, *TargetLength*, *TargetBuffer*, *DataLength*, *CompCode*, *Reason*)


### Parâmetros

#### Hconn

Tipo: MQHCONN - entrada

Esta manipulação representa a conexão ao gerenciador de filas.

Em uma saída de conversão de dados, Hconn é normalmente o identificador que é transmitido para a saída de conversão de dados no campo Hconn da estrutura MQDXP; esse identificador não é necessariamente o mesmo identificador especificado pelo aplicativo que emitiu a chamada MQGET.

 Em IBM i, o valor especial a seguir pode ser especificado para Hconn:

#### MQHC\_DEF\_HCONN

Manipulação de conexões padrão.

Se você executar um aplicativo CICS TS 3.2 ou superior, assegure que o programa de saída de conversão de caracteres, que chama a chamada MQXCNCV, esteja definido como OPENAPI. Essa definição evita o erro 2018 MQRC\_HCONN\_ERROR causado por uma conexão incorreta e permite que o MQGET seja concluído.

#### Opções

Tipo: MQLONG - entrada

Opções que controlam a ação de MQXCNCV.

Zero ou mais opções descritas podem ser especificadas. Para especificar mais de uma opção, inclua os valores juntos (não inclua a mesma constante mais de uma vez) ou combine os valores usando a operação OR bit a bit (se a linguagem de programação suportar operações de bit).

**Opção de conversão padrão:** a opção a seguir controla o uso da conversão de caracteres padrão..

#### MQDCC\_DEFAULT\_CONVERSION

Conversão padrão..

Esta opção especifica que a conversão de caracteres padrão pode ser utilizada se um ou ambos os conjuntos de caracteres especificados na chamada não forem suportados. Isso permite que o gerenciador de filas use um conjunto de caracteres padrão especificado pela instalação que se aproxima do conjunto de caracteres especificado ao converter a sequência.

**Nota:** O resultado do uso de um conjunto de caracteres aproximado para converter a sequência é que alguns caracteres podem ser convertidos incorretamente. Isso pode ser evitado usando na sequência apenas caracteres que são comuns ao conjunto de caracteres especificado e ao conjunto de caracteres padrão.

Os conjuntos de caracteres padrão são definidos por uma opção de Configuração quando o gerenciador de filas é instalado ou reiniciado.

Se MQDCC\_DEFAULT\_CONVERSION não for especificado, o gerenciador de filas usará apenas os conjuntos de caracteres especificados para converter a cadeia e a chamada falhará se um ou ambos os conjuntos de caracteres não for suportado.

Essa opção é suportada nos seguintes ambientes:

-  AIX
-  IBM i
-  Linux
-  Windows

**Opção de preenchimento:** A opção a seguir permite que o gerenciador de filas preencha a sequência convertida com espaços em branco ou descarte caracteres finais insignificantes, para que a sequência convertida se ajuste ao buffer de destino:

### **MQDCC\_FILL\_TARGET\_BUFFER**

Preencha o buffer de destino

Esta opção solicita que a conversão ocorra de forma que o buffer de destino seja preenchido completamente:

- Se a sequência se contrai quando é convertida, espaços em branco à direita são incluídos para preencher o buffer de destino.
- Se a sequência for expandida quando for convertida, os caracteres finais que não forem significativos serão descartados para que a sequência convertida se ajuste ao buffer de destino. Se isso puder ser feito com êxito, a chamada será concluída com MQCC\_OK e o código de razão MQRC\_NONE.

Se houver poucos caracteres finais insignificantes, a maior parte da sequência que puder caber será colocada no buffer de destino e a chamada será concluída com MQCC\_WARNING e o código de razão MQRC\_CONVERTED\_MSG\_TOO\_BIG.

Caracteres insignificantes são:

- Espaços em branco finais
- Caracteres após o primeiro caractere nulo na cadeia (mas excluindo o primeiro caractere nulo em si)
- Se a sequência, `TargetCCSID` e `TargetLength` forem tais que o buffer de destino não possa ser configurado completamente com caracteres válidos, a chamada falhará com MQCC\_FAILED e o código de razão MQRC\_TARGET\_LENGTH\_ERROR. Isso pode ocorrer quando `TargetCCSID` é um conjunto de caracteres DBCS puro (como UTF-16), mas `TargetLength` especifica um comprimento que é um número ímpar de bytes.
- `TargetLength` pode ser menor que ou maior que `SourceLength`. No retorno de MQXCNCV, `DataLength` tem o mesmo valor de `TargetLength`.

Se esta opção não for especificada:

- A sequência tem permissão para contrair ou expandir dentro do buffer de destino, conforme necessário. Caracteres finais insignificantes não são incluídos ou descartados.

Se a sequência convertida se ajustar no buffer de destino, a chamada será concluída com MQCC\_OK e o código de razão MQRC\_NONE

Se a sequência convertida for muito grande para o buffer de destino, a quantidade da sequência que se ajustar será colocada no buffer de destino e a chamada será concluída com MQCC\_WARNING e o código de razão MQRC\_CONVERTED\_MSG\_TOO\_BIG. Nesse caso, é possível retornar menos de `TargetLength` bytes.

- `TargetLength` pode ser menor que ou maior que `SourceLength`. No retorno de MQXCNCV, `DataLength` é menor ou igual a `TargetLength`.

Essa opção é suportada nos seguintes ambientes:

-  AIX
-  IBM i
-  Linux
-  Windows

**Opções de codificação:** As opções descritas podem ser usadas para especificar as codificações de número inteiro das sequências de origem e destino. A codificação relevante é usada apenas quando o identificador do conjunto de caracteres correspondente indica que a representação do conjunto de caracteres no armazenamento principal depende da codificação usada para números inteiros binários. Isso afeta apenas alguns conjuntos de caracteres multibyte (por exemplo, conjuntos de caracteres UTF-16).

A codificação será ignorada se o conjunto de caracteres for um conjunto de caracteres de byte único (SBCS), ou um conjunto de caracteres multibyte com representação no armazenamento principal que não depende da codificação de número inteiro.

Apenas um dos valores MQDCC\_SOURCE\_\* deve ser especificado, combinado com um dos valores MQDCC\_TARGET\_\*:

**MQDCC\_SOURCE\_ENC\_NATIVE**

Codificação de origem é o padrão para o ambiente e a linguagem de programação.

**MQDCC\_SOURCE\_ENC\_NORMAL**

A codificação de origem é normal

**MQDCC\_SOURCE\_ENC\_REVERTIDO**

A codificação de origem é revertida

**MQDCC\_SOURCE\_ENC\_UNDEFINED**

A codificação de origem é indefinida

**MQDCC\_TARGET\_ENC\_NATIVE**

A codificação de destino é o padrão para o ambiente e linguagem de programação.

**MQDCC\_TARGET\_ENC\_NORMAL**

A codificação de destino é normal

**MQDCC\_TARGET\_ENC\_REVERTIDO**

A codificação de destino é revertida

**MQDCC\_TARGET\_ENC\_UNDEFINED**

A codificação de destino é indefinida

Os valores de codificação definidos anteriormente podem ser incluídos diretamente no campo `Options`. No entanto, se a codificação de origem ou de destino for obtida do campo `Encoding` no MQMD ou em outra estrutura, o processamento a seguir deverá ser feito:

1. A codificação de número inteiro deve ser extraída do campo `Encoding` eliminando as codificações flutuantes e decimais compactadas; consulte [“Analisando codificações” na página 934](#) para obter detalhes de como fazer isso.
2. A codificação de número inteiro resultante da etapa 1 deve ser multiplicada pelo fator apropriado antes de ser incluída no campo `Options`. Esses fatores são:
  - MQDCC\_SOURCE\_ENC\_FACTOR para a codificação de origem
  - MQDCC\_TARGET\_ENC\_FACTOR para a codificação de destino

O código de exemplo a seguir ilustra como isso pode ser codificado na linguagem de programação C:

```
Options = (MsgDesc.Encoding & MQENC_INTEGER_MASK)
          * MQDCC_SOURCE_ENC_FACTOR
```

```
+ (DataConvExitParms.Encoding & MQENC_INTEGER_MASK)
* MQDCC_TARGET_ENC_FACTOR;
```

Se não especificado, as opções de codificação serão padronizadas como indefinidas (MQDCC\_\*\_ENC\_UNDEFINED). Na maioria dos casos, isso não afeta a conclusão bem-sucedida da chamada MQXCNVC.. No entanto, se o conjunto de caracteres correspondente for um conjunto de caracteres multibyte com representação dependente da codificação (por exemplo, um conjunto de caracteres UTF-16 ), a chamada falhará com o código de razão MQRC\_SOURCE\_INTEGER\_ENC\_ERROR ou MQRC\_TARGET\_INTEGER\_ENC\_ERROR conforme apropriado.

As opções de codificação são suportadas nos seguintes ambientes:

-  AIX
-  IBM i
-  Linux
-  Windows
-  z/OS

**Opção padrão:** Se nenhuma das opções descritas anteriormente for especificada a seguinte opção pode ser usada:

#### **MQDCC\_NONE**

Nenhuma opção especificada.

MQDCC\_NONE é definido para a documentação do programa de auxílio Não se pretende que essa opção seja usada com nenhuma outra, mas como seu valor é zero, tal uso não pode ser detectado.

#### **SourceCCSID**

Tipo: MQLONG - entrada

Esse é o identificador de conjunto de caracteres codificados da sequência de entrada em SourceBuffer.

#### **SourceLength**

Tipo: MQLONG - entrada

Esse é o comprimento em bytes da sequência de entrada em SourceBuffer ; deve ser zero ou maior.

#### **SourceBuffer**

Tipo: MQCHAR x SourceLength -entrada

Este é o buffer que contém a cadeia a ser convertida de um conjunto de caracteres para outro

#### **TargetCCSID**

Tipo: MQLONG - entrada

Este é o identificador do conjunto de caractere codificado do conjunto de caracteres para o qual SourceBuffer deve ser convertido.

#### **TargetLength**

Tipo: MQLONG - entrada

Esse é o comprimento em bytes do buffer de saída TargetBuffer ; deve ser zero ou maior. Pode ser menor ou maior que SourceLength.

#### **TargetBuffer**

Tipo: MQCHAR x TargetLength -saída

Esta é a sequência após ela ter sido convertida para o conjunto de caracteres definido por TargetCCSID A sequência convertida pode ser menor ou maior que a sequência não convertida. O parâmetro **DataLength** indica o número de bytes válidos retornados

**DataLength**

Tipo: MQLONG - saída

Este é o comprimento da sequência retornada no buffer de saída TargetBuffer. A sequência convertida pode ser menor ou maior que a sequência não convertida.

**CompCode**

Tipo: MQLONG - saída

É um dos seguintes:

**MQCC\_OK**

Indica conclusão bem-sucedida.

**MQCC\_WARNING**

Aviso (conclusão parcial).

**MQCC\_FAILED**

A chamada falhou.

**Razão**

Tipo: MQLONG - saída

Código de razão qualificando CompCode.

Se CompCode for MQCC\_OK:

**MQRC\_NONE**

(0, X'000') Nenhuma razão a relatar.

Se CompCode for MQCC\_WARNING:

**MQRC\_CONVERTED\_MSG\_TOO\_BIG**

(2120, X'848') Dados convertidos muito grandes para o buffer.

Se CompCode for MQCC\_FAILED:

**MQRC\_DATA\_LENGTH\_ERROR**

(2010, X'7DA') Parâmetro de comprimento de dados inválido.

**MQRC\_DBCS\_ERROR**

(2150, X'866') Sequência DBCS inválida.

**MQRC\_HCONN\_ERROR**

(2018, X'7E2') Manipulação de conexões não válida.

**MQRC\_OPTIONS\_ERROR**

(2046, X'7FE') Opções não válidas ou não consistentes.

**MQRC\_RESOURCE\_PROBLEM**

(2102, X'836') Recursos insuficientes do sistema disponíveis.

**MQRC\_SOURCE\_BUFFER\_ERROR**

(2145, X'861') Parâmetro de buffer de origem inválido.

**MQRC\_SOURCE\_CCSID\_ERROR**

(2111, X'83F') Identificador do conjunto de caracteres codificados da origem inválido.

**MQRC\_SOURCE\_INTEGER\_ENC\_ERROR**

(2112, X'840') Codificação de número inteiro da origem não reconhecida.

**MQRC\_SOURCE\_LENGTH\_ERROR**

(2143, X'85F') Parâmetro de comprimento de origem inválido.

**MQRC\_STORAGE\_NOT\_AVAILABLE**

(2071, X'817') Armazenamento insuficiente disponível.

**MQRC\_TARGET\_BUFFER\_ERROR**

(2146, X'862') Parâmetro de buffer de destino inválido.

**MQRC\_TARGET\_CCSID\_ERROR**

(2115, X'843') Identificador do conjunto de caracteres codificados do destino inválido.



### **MQRC\_TARGET\_INTEGER\_ENC\_ERROR**

(2116, X'844') Codificação de número inteiro do destino não reconhecida.

### **MQRC\_TARGET\_LENGTH\_ERROR**

(2144, X'860 ') O parâmetro de comprimento de destino não é válido.

### **MQRC\_UNEXPECTED\_ERROR**

(2195, X'893') Ocorreu um erro inesperado.

Para obter informações detalhadas sobre esses códigos, consulte [Mensagens e códigos de razão](#).

## **Chamada C**

```
MQXCNCV (Hconn, Options, SourceCCSID, SourceLength, SourceBuffer,  
         TargetCCSID, TargetLength, TargetBuffer, &DataLength,  
         &CompCode, &Reason);
```

Declare os parâmetros conforme a seguir:

```
MQHCONN  Hconn;           /* Connection handle */  
MQLONG   Options;        /* Options that control the action of  
                        MQXCNCV */  
MQLONG   SourceCCSID;    /* Coded character set identifier of string  
                        before conversion */  
MQLONG   SourceLength;   /* Length of string before conversion */  
MQCHAR   SourceBuffer[n]; /* String to be converted */  
MQLONG   TargetCCSID;    /* Coded character set identifier of string  
                        after conversion */  
MQLONG   TargetLength;   /* Length of output buffer */  
MQCHAR   TargetBuffer[n]; /* String after conversion */  
MQLONG   DataLength;     /* Length of output string */  
MQLONG   CompCode;       /* Completion code */  
MQLONG   Reason;         /* Reason code qualifying CompCode */
```

## **Declaração COBOL (somente IBM i)**

IBM i

```
CALL 'MQXCNCV' USING HCONN, OPTIONS, SOURCECCSID, SOURCELENGTH,  
                    SOURCEBUFFER, TARGETCCSID, TARGETLENGTH,  
                    TARGETBUFFER, DATALENGTH, COMPCODE, REASON.
```

Declare os parâmetros conforme a seguir:

```
** Connection handle  
01 HCONN          PIC S9(9) BINARY.  
** Options that control the action of MQXCNCV  
01 OPTIONS        PIC S9(9) BINARY.  
** Coded character set identifier of string before conversion  
01 SOURCECCSID   PIC S9(9) BINARY.  
** Length of string before conversion  
01 SOURCELENGTH  PIC S9(9) BINARY.  
** String to be converted  
01 SOURCEBUFFER   PIC X(n).  
** Coded character set identifier of string after conversion  
01 TARGETCCSID   PIC S9(9) BINARY.  
** Length of output buffer  
01 TARGETLENGTH  PIC S9(9) BINARY.  
** String after conversion  
01 TARGETBUFFER  PIC X(n).  
** Length of output string  
01 DATALENGTH   PIC S9(9) BINARY.  
** Completion code  
01 COMPCODE      PIC S9(9) BINARY.  
** Reason code qualifying COMPCODE  
01 REASON        PIC S9(9) BINARY.
```

## Declaração do assembler S/390

```
CALL MQXCNCV, (HCONN, OPTIONS, SOURCECCSID, SOURCELENGTH,          X
              SOURCEBUFFER, TARGETCCSID, TARGETLENGTH, TARGETBUFFER, X
              DATALENGTH, COMPCODE, REASON)
```

Declare os parâmetros conforme a seguir:

HCONN	DS	F	Connection handle
OPTIONS	DS	F	Options that control the action of MQXCNCV
SOURCECCSID	DS	F	Coded character set identifier of string before *
SOURCELENGTH	DS	F	Length of string before conversion
SOURCEBUFFER	DS	CL(n)	String to be converted
TARGETCCSID	DS	F	Coded character set identifier of string after *
TARGETLENGTH	DS	F	Length of output buffer
TARGETBUFFER	DS	CL(n)	String after conversion
DATALENGTH	DS	F	Length of output string
COMPCODE	DS	F	Completion code
REASON	DS	F	Reason code qualifying COMPCODE

## MQ\_DATA\_CONV\_EXIT-Saída de conversão de dados

A chamada MQ\_DATA\_CONV\_EXIT descreve os parâmetros que são transmitidos para a saída de conversão de dados

Nenhum ponto de entrada chamado MQ\_DATA\_CONV\_EXIT é fornecido pelo gerenciador de filas (consulte a nota de uso [11](#) )

Esta definição faz parte da IBM MQ Data Conversion Interface (DCI), que é uma das interfaces de estrutura do IBM MQ .

### Sintaxe

MQ\_DATA\_CONV\_EXIT (*DataConvExitParms*, *MsgDesc*, *InBufferComprimento*, *InBuffer*, *OutBufferComprimento*, *OutBuffer*)

### Parâmetros

#### DataConvExitParms

Tipo: MQDXP-entrada/saída

Essa estrutura contém informações relacionadas à invocação da saída A saída configura as informações nessa estrutura para indicar o resultado da conversão Consulte [“MQDXP-Parâmetro de saída de conversão de dados”](#) na página 946 para obter detalhes dos campos nesta estrutura..

#### MsgDesc

Tipo: MQMD - entrada/saída

Na entrada para a saída, este é o descritor de mensagens associado com os dados da mensagem transmitidos para a saída no parâmetro **InBuffer**

**Nota:** O parâmetro **MsgDesc** passado para a saída é sempre a versão mais recente do MQMD suportada pelo gerenciador de fila que chama a saída Se a saída for destinada a ser móvel entre ambientes diferentes, a saída verificará o campo **Version** em **MsgDesc** para verificar se os campos que a saída precisa acessar estão presentes na estrutura..

Nos ambientes a seguir, a saída é transmitida um MQMD version-2 :

-  AIX
-  IBM i
-  Linux

- **Windows** Windows

Em todos os outros ambientes que suportam a saída de conversão de dados, a saída é transmitida um MQMD version-1 .

Na saída, a saída alterará os campos `Encoding` e `CodedCharSetId` para os valores solicitados pelo aplicativo, se a conversão tiver sido bem-sucedida; essas mudanças serão refletidas de volta para o aplicativo. Quaisquer outras mudanças que a saída faz na estrutura são ignoradas; elas não são refletidas de volta para o aplicativo.

Se a saída retornar `MQXDR_OK` no campo `ExitResponse` da estrutura `MQDXP`, mas não alterar os campos `Encoding` ou `CodedCharSetId` no descritor de mensagem, o gerenciador de filas retornará para esses campos os valores que os campos correspondentes na estrutura `MQDXP` tinham na entrada para a saída.

### **Comprimento do InBuffer**

Tipo: `MQLONG` - entrada

Comprimento em bytes de `InBuffer`.

Esse é o comprimento do buffer de entrada `InBuffer` e especifica o número de bytes a serem processados pela saída. `InBufferLength` é o menor comprimento dos dados da mensagem antes da conversão e o comprimento do buffer fornecido pelo aplicativo na chamada `MQGET`.

O valor é sempre maior do que zero.

### **InBuffer**

Tipo: `MQBYTEInBufferLength` - entrada

Buffer contendo a mensagem não convertida.

Isso contém os dados da mensagem antes da conversão. Se a saída não puder converter os dados, o gerenciador de filas retornará o conteúdo desse buffer para o aplicativo após a conclusão da saída.

**Nota:** A saída não deve alterar `InBuffer`; Se esse parâmetro for alterado, os resultados serão indefinidos.

Na linguagem de programação C, esse parâmetro é definido como um ponteiro para vazio.

### **Comprimento do OutBuffer**

Tipo: `MQLONG` - entrada

Comprimento em bytes de `OutBuffer`.

Esse é o comprimento do buffer de saída `OutBuffer` e é o mesmo comprimento do buffer fornecido pelo aplicativo na chamada `MQGET`.

O valor é sempre maior do que zero.

### **OutBuffer**

Tipo: `MQBYTEOutBufferLength` - saída

Buffer contendo a mensagem convertida.

Na saída da saída, se a conversão foi bem-sucedida (conforme indicado pelo valor `MQXDR_OK` no campo `ExitResponse` do parâmetro **DataConvExitParms**), `OutBuffer` contém os dados da mensagem a serem entregues para o aplicativo, na representação solicitada. Se a conversão foi malsucedida, quaisquer alterações que a saída tenha feito para este buffer serão ignorados.

Na linguagem de programação C, esse parâmetro é definido como um ponteiro para vazio.

## **Observações de Uso**

1. Uma saída de conversão de dados é uma saída gravada pelo usuário que recebe o controle durante o processamento de uma chamada `MQGET`. A função executada pela saída de conversão de dados é definida pelo provedor da saída; no entanto, a saída deve estar em conformidade com as regras descritas aqui e na estrutura de parâmetro `MQDXP` associada.

As linguagens de programação que podem ser usadas para uma saída de conversão de dados são determinadas pelo ambiente.

2. A saída será chamada apenas se todas as seguintes instruções forem verdadeiras:

- A opção MQGMO\_CONVERT é especificado na chamada MQGET..
- O campo Format no descritor de mensagens não é MQFMT\_NONE
- A mensagem ainda não está na representação necessária; ou seja, um ou ambos os CodedCharSetId e Encoding da mensagem são diferentes do valor especificado pelo aplicativo no descritor de mensagens fornecido na chamada MQGET
- O gerenciador de filas ainda não fez a conversão com êxito
- O comprimento do buffer do aplicativo é maior que zero
- O comprimento dos dados da mensagem é maior que zero
- O código de razão até agora durante a operação MQGET é MQRC\_NONE ou MQRC\_TRUNCATED\_MSG\_ACCEPTED

3. Quando uma saída estiver sendo gravada, considere a codificação da saída de uma maneira que permita converter mensagens que foram truncadas. Mensagens truncadas podem surgir das seguintes maneiras:

- O aplicativo de recebimento fornece um buffer menor que a mensagem, mas especifica a opção MQGMO\_ACCEPT\_TRUNCATED\_MSG na chamada MQGET.

Nesse caso, o campo Reason no parâmetro **DataConvExitParms** na entrada para a saída tem o valor MQRC\_TRUNCATED\_MSG\_ACCEPTED..

- O emissor da mensagem a truncou antes de enviá-la. Isso pode acontecer com mensagens de relatório, por exemplo (consulte [“Conversão de mensagens de relatório” na página 945](#) para obter mais detalhes)

Nesse caso, o campo Reason no parâmetro **DataConvExitParms** na entrada para a saída tem o valor MQRC\_NONE (se o aplicativo de recebimento forneceu um buffer que era grande o suficiente para a mensagem)..

Portanto, o valor do campo Reason na entrada para a saída nem sempre pode ser usado para decidir se a mensagem foi truncada.

A característica distintiva de uma mensagem truncada é que o comprimento fornecido para a saída no parâmetro **InBufferLength** é inferior ao comprimento implícito pelo nome do formato contido no campo Format no descritor de mensagens. Portanto, a saída deve verificar o valor de InBufferLength antes de tentar converter qualquer um dos dados; a saída não deve assumir que a quantidade total de dados implícita pelo nome do formato foi fornecida.

Se a saída não tiver sido gravada para converter mensagens truncadas e InBufferLength for menor que o valor esperado, a saída retornará MQXDR\_CONVERSION\_FAILED no campo ExitResponse do parâmetro **DataConvExitParms**, com os campos CompCode e Reason configurados como MQCC\_WARNING e MQRC\_FORMAT\_ERROR.

Se a saída tiver sido gravada para converter mensagens truncadas, a saída converterá o máximo de dados possível (consulte a próxima nota de uso), tomando cuidado para não tentar examinar ou converter dados além do final de InBuffer.. Se a conversão for concluída com sucesso, a saída deixará o campo Reason no parâmetro **DataConvExitParms** inalterado. Isso retorna MQRC\_TRUNCATED\_MSG\_ACCEPTED se a mensagem foi truncada pelo gerenciador de filas do receptor e MQRC\_NONE se a mensagem foi truncada pelo emissor da mensagem.

Também é possível expandir uma mensagem durante a conversão, para o ponto em que ela é maior que OutBuffer. Nesse caso, a saída deve decidir se truncar a mensagem; o campo AppOptions no parâmetro **DataConvExitParms** indica se o aplicativo de recebimento especificou a opção MQGMO\_ACCEPT\_TRUNCATED\_MSG.

4. Geralmente, todos os dados na mensagem fornecidos para a saída em InBuffer são convertidos, ou nenhum deles é. Uma exceção a isso, no entanto, ocorrerá se a mensagem for truncada, antes da conversão ou durante a conversão; neste caso, pode haver um item incompleto no final do buffer (por

exemplo: 1 byte de um caractere de byte duplo ou 3 bytes de um número inteiro de 4 bytes). Nessa situação, considere omitir o item incompleto e configurar os bytes não utilizados no OutBuffer para nulos. No entanto, elementos ou caracteres completos dentro de uma matriz ou sequência devem ser convertidos.


5. Quando uma saída for necessária pela primeira vez, o gerenciador de filas tentará carregar um objeto que tenha o mesmo nome que o formato (além de extensões). O objeto carregado deve conter a saída que processa mensagens com esse nome de formato.. Considere tornar o nome da saída e o nome do objeto que contém a saída idêntica, embora nem todos os ambientes requeiram isso
6. Uma nova cópia da saída é carregada quando um aplicativo tenta recuperar a primeira mensagem que usa esse Format desde o aplicativo conectado ao gerenciador de filas. Para aplicativos CICS ou IMS , isto significa quando o subsistema CICS ou IMS está conectado ao gerenciador de filas. Uma nova cópia também pode ser carregada em outros momentos, se o gerenciador de fila tiver descartado uma cópia carregada anteriormente. Por essa razão, uma saída não deve tentar usar armazenamento estático para comunicar informações de uma chamada da saída para a próxima-a saída pode ser descarregada entre as duas chamadas.
7. Se houver uma saída fornecida pelo usuário com o mesmo nome que um dos formatos integrados suportados pelo gerenciador de fila, a saída fornecida pelo usuário não substituirá a rotina de conversão integrada. As únicas circunstâncias em que tal saída é invocada são:
  - Se a rotina de conversão integrada não puder manipular conversões para ou a partir do CodedCharSetId ou Encoding envolvido, ou
  - Se a rotina de conversão integrada falhou ao converter os dados (por exemplo, porque há um campo ou caractere que não pode ser convertido).
8. O escopo da saída é dependente do ambiente. Formatos nomes devem ser escolhidos para minimizar o risco de conflitos com outros formatos. Considere iniciar com caracteres que identifiquem o aplicativo que define o nome do formato.
9. A saída de conversão de dados é executada em um ambiente como aquele do programa que emitiu a chamada MQGET; o ambiente inclui espaço de endereço e perfil do usuário (onde aplicável). O programa pode ser um agente do canal de mensagens enviando mensagens para um gerenciador de filas de destino que não suporta conversão de mensagens. A saída não pode comprometer a integridade do gerenciador de filas, pois ele não é executado no ambiente do gerenciador de filas.
10. A única chamada MQI que pode ser usada pela saída é MQXCNV; a tentativa de usar outras chamadas MQI falha com código de razão MQRC\_CALL\_IN\_PROGRESS ou outros erros imprevisíveis.
11. Nenhum ponto de entrada chamado MQ\_DATA\_CONV\_EXIT é fornecido pelo gerenciador da fila. No entanto, um typedef é fornecido para o nome MQ\_DATA\_CONV\_EXIT na linguagem de programação C e isso pode ser usado para declarar a saída gravada pelo usuário, para assegurar que os parâmetros estejam corretos. O nome da saída deve ser igual ao nome do formato (o nome contido no campo Format em MQMD), embora isso não seja necessário em todos os ambientes.

O exemplo a seguir ilustra como a saída que processa o formato MYFORMAT pode ser declarada na linguagem de programação C:

```
#include "cmqc.h"
#include "cmqxc.h"

MQ_DATA_CONV_EXIT MYFORMAT;

void MQENTRY MYFORMAT(
    PMQDXP pDataConvExitParms, /* Data-conversion exit parameter
                               block */
    PMQMD pMsgDesc,           /* Message descriptor */
    MQLONG InBufferLength,    /* Length in bytes of InBuffer */
    PMQVOID pInBuffer,       /* Buffer containing the unconverted
                               message */
    MQLONG OutBufferLength,   /* Length in bytes of OutBuffer */
    PMQVOID pOutBuffer)      /* Buffer containing the converted
                               message */
{
    /* C language statements to convert message */
}
```

12.  No z/OS, se uma saída cruzada da API também estiver em vigor, ela será chamada após a saída de conversão de dados.

## Chamada C

```
exitname (&DataConvExitParms, &MsgDesc, InBufferLength,  
         InBuffer, OutBufferLength, OutBuffer);
```

Os parâmetros transmitidos para a saída são declarados da seguinte forma:

```
MQDXP  DataConvExitParms; /* Data-conversion exit parameter block */  
MQMD   MsgDesc;          /* Message descriptor */  
MQLONG InBufferLength;   /* Length in bytes of InBuffer */  
MQBYTE InBuffer[n];     /* Buffer containing the unconverted  
                        message */  
MQLONG OutBufferLength; /* Length in bytes of OutBuffer */  
MQBYTE OutBuffer[n];    /* Buffer containing the converted  
                        message */
```

## Declaração COBOL (somente IBM i)

 IBM i

```
CALL 'exitname' USING DATACONVEXITPARMS, MSGDESC, INBUFFERLENGTH,  
                     INBUFFER, OUTBUFFERLENGTH, OUTBUFFER.
```

Os parâmetros transmitidos para a saída são declarados da seguinte forma:

```
** Data-conversion exit parameter block  
01 DATACONVEXITPARMS.  
   COPY CMQDXPV.  
** Message descriptor  
01 MSGDESC.  
   COPY CMQMDV.  
** Length in bytes of INBUFFER  
01 INBUFFERLENGTH PIC S9(9) BINARY.  
** Buffer containing the unconverted message  
01 INBUFFER PIC X(n).  
** Length in bytes of OUTBUFFER  
01 OUTBUFFERLENGTH PIC S9(9) BINARY.  
** Buffer containing the converted message  
01 OUTBUFFER PIC X(n).
```

## Declaração do assembler System/390

```
CALL EXITNAME,(DATACONVEXITPARMS,MSGDESC,INBUFFERLENGTH,      X  
               INBUFFER,OUTBUFFERLENGTH,OUTBUFFER)
```

Os parâmetros transmitidos para a saída são declarados da seguinte forma:

```
DATACONVEXITPARMS CMQDXPA , Data-conversion exit parameter block  
MSGDESC           CMQMDA , Message descriptor  
INBUFFERLENGTH   DS      F Length in bytes of INBUFFER  
INBUFFER         DS      CL(n) Buffer containing the unconverted  
*               * message  
OUTBUFFERLENGTH  DS      F Length in bytes of OUTBUFFER  
OUTBUFFER        DS      CL(n) Buffer containing the converted  
*               * message
```

## Propriedades especificadas como elementos MQRFH2

As propriedades do descritor de não mensagens podem ser especificadas como elementos nas pastas de cabeçalho MQRFH2 . Visão geral de elementos MQRFH2 sendo especificados como propriedades.

Isso retém a compatibilidade com as versões anteriores dos clientes IBM MQ JMS e XMS . Esta seção descreve como especificar propriedades em cabeçalho MQRFH2 .

Para usar elementos MQRFH2 como propriedades, especifique os elementos conforme descrito em [Usando IBM MQ classes for Java](#) . Essas informações complementam as informações descritas em [“MQRFH2 - Regras e Formatação do Cabeçalho 2”](#) na página 547

## Mapeando tipos de dados de propriedade para tipos de dados MQRFH2

Este tópico fornece informações sobre os tipos de propriedade de mensagem mapeados para seus tipos de dados MQRFH2 correspondentes.

Tipo propriedade mensagem	Tipo de dados MQRFH2
MQBYTE []	bin.hex
MQBOOL	booleano
MQINT8	i1
MQINT16	i2
MQINT32	i4
MQINT64	i8
MQFLOAT32	r4
MQFLOAT64	r8
MQCHAR []	sequência

Qualquer elemento sem um tipo de dados é assumido como sendo do tipo "string".

Um tipo de dados MQRFH2 de int, que significa um número inteiro de tamanho não especificado, é tratado como se fosse um i8.

Um valor nulo é indicado pelo atributo do elemento `xsi:nil='true'` Não use o atributo `xsi:nil='false'` para valores não nulos

Por exemplo, a propriedade a seguir possui um valor nulo:

```
<NullProperty xsi:nil='true'></NullProperty>
```

Uma propriedade de cadeia de bytes ou caracteres pode ter um valor vazio. Isso é representado por um elemento MQRFH2 com um valor de elemento de comprimento zero..

Por exemplo, a propriedade a seguir possui um valor vazio:

```
<EmptyProperty></EmptyProperty>
```

## Pastas MQRFH2 suportadas

Visão Geral do Uso de Campos do Descritor de Mensagens como Propriedades

As pastas `<jms>`, `<mcd>`, `<mqext>` e `<usr>` são descritas em O cabeçalho MQRFH2 e JMS. A pasta `<usr>` é usada para transportar quaisquer propriedades definidas pelo aplicativo JMS associadas a uma mensagem. Os grupos não são permitidos na pasta `<usr>`

O cabeçalho MQRFH2 e JMS suportam as seguintes pastas adicionais:

- <mq>

Essa pasta é usada para propriedades definidas pelo MQ que são usadas pelo IBM MQ.

- <mq\_usr>

Essa pasta pode ser usada para transportar quaisquer propriedades definidas pelo aplicativo que não sejam expostas como propriedades definidas pelo usuário do JMS, pois as propriedades podem não atender aos requisitos de uma propriedade JMS. Essa pasta pode conter grupos que a pasta <usr> não pode conter.

- Qualquer pasta marcada com o atributo `content='properties'`.

Tal pasta é equivalente à pasta <mq\_usr> em conteúdo.

- <mmps>

Essa pasta é usada para propriedades de publicação / assinatura IBM MQ.

O IBM MQ também suporta as seguintes pastas que já estão em uso pelo WAS/SIB:

- <sib>

Essa pasta é usada e reservada para propriedades de mensagem do sistema WAS/SIB que não são expostas como propriedades JMS ou são mapeadas para as propriedades JMS\_IBM\_\*, mas são expostas para aplicativos WAS/SIB; elas incluem propriedades de caminhos de roteamento de encaminhamento e reverso.

Pelo menos alguns não podem ser expostos como propriedades JMS, porque eles são matrizes de bytes. Se seu aplicativo incluir propriedades nessa pasta, o valor será ignorado ou removido.

- <sib\_usr>

Essa pasta é usada e reservada para propriedades de mensagem do usuário WAS/SIB que não podem ser expostas como propriedades do usuário JMS porque elas não são de tipos suportados; elas são expostas para aplicativos WAS/SIB.

Essas são propriedades do usuário, que você pode obter ou configurar por meio da interface SIMessage, mas o conteúdo da matriz de bytes é mapeado para o valor da propriedade necessário.

Se seu aplicativo IBM MQ gravar um elemento `bin.hex` arbitrário na pasta, o aplicativo provavelmente receberá um `IOException`, pois ele não é do formato esperado para restauração. Se você incluir algo diferente de um elemento `bin.hex`, receberá um `ClassCastException`.

Não tente disponibilizar propriedades para WAS/SIB usando essa pasta; em vez disso, use a pasta <usr> para esse propósito.

- <sib\_context>

Esta pasta é usada para as propriedades de mensagem do sistema WAS/SIB que não são expostas a aplicativos de usuário WAS/SIB ou como propriedades JMS. Elas incluem propriedades de segurança e transacionais que são usadas para serviços da Web e semelhantes.

Seu aplicativo não deve incluir propriedades nesta pasta..

- <mqema>

Essa pasta foi usada por WAS/SIB em vez de pela pasta <mqext>

Os nomes da pasta MQRFH2 fazem distinção entre maiúsculas e minúsculas.

As seguintes pastas são reservadas, em qualquer combinação de caracteres minúsculos ou maiúsculos:

- Qualquer pasta prefixada por `mq` ou `wmq`, reservado para uso por IBM MQ..
- Qualquer pasta prefixada por `sib`; reservado para uso por WAS/SIB.
- Pastas <Root> e <Body>; reservado mas não usado.

As pastas a seguir não são reconhecidas como contendo propriedades de mensagem:



- <psc>

Usado pelo IBM Integration Bus para transmitir mensagens de comando de publicação / assinatura para o broker

- <pscri>

Usado pelo IBM Integration Bus para conter informações do broker, em resposta às mensagens de comando de publicação / assinatura

- Qualquer pasta não definida por IBM, que não esteja marcada com o atributo `content='properties'`

Não especifique `content='properties'` nas pastas <psc> ou <pscri> .. Se você fizer isso, essas pastas serão tratadas como propriedades e IBM Integration Bus provavelmente parará de funcionar conforme esperado.

Se seu aplicativo estiver construindo mensagens com propriedades, nos cabeçalhos MQRFH2 a serem reconhecidos como um cabeçalho MQRFH2 contendo propriedades, o cabeçalho deverá estar na lista de cabeçalhos que podem ser encadeados no cabeçalho da mensagem..

O MQRFH2 pode ser precedido por qualquer número de cabeçalhos padrão MQH, ou um MQCIH, um MQDLH, um MQIIH, um MQTM, um MQTMC2 ou um MQXQH. Uma sequência ou um MQCFH termina a análise porque eles não podem ser encadeadas

É possível que uma mensagem contenha vários cabeçalhos MQRFH2, todos com propriedades de mensagem. Pastas com o mesmo nome podem coexistir em cabeçalhos diferentes, a menos que restritos de outra forma, por exemplo, WAS/SIB. As pastas são tratadas como uma pasta lógica, se todas elas estiverem em cabeçalhos significativos

Enquanto as pastas dos cabeçalhos significativos não podem ser mescladas com essas pastas em cabeçalhos não significativos, as pastas com o mesmo nome dentro dos cabeçalhos significativos podem ser mescladas, removendo quaisquer propriedades conflitantes. Seus aplicativos não devem depender do layout de propriedades dentro de suas mensagens.

Os grupos MQRFH2 são analisados para propriedades em pastas definidas pelo usuário, ou seja, não <wmq>, <jms>, <mcd>, <usr>, <mqext>, <sib>, <sib\_usr>, <sib\_context> e <mqema> pastas.

Os grupos nas pastas de propriedades definidas pelo IBM, exceto para as pastas <wmq> e <mq>, são analisados para propriedades

Uma pasta MQRFH2 não pode conter conteúdo misto; uma pasta ou grupo pode conter grupos ou propriedades ou um valor, mas não ambos.

Um segmento de uma mensagem, o primeiro segmento ou um segmento subsequente, não pode conter propriedades definidas por IBM MQ além daquelas no descritor de mensagens. Portanto, colocar uma mensagem contendo essas propriedades com o conjunto MQMF\_SEGMENT ou MQMF\_SEGMENTATION\_ALLOWED faz com que o put falhe com MQRC\_SEGMENTATION\_NOT\_ALLOWED


No entanto, os grupos de mensagens podem conter propriedades definidas por IBM MQ

## Geração de cabeçalhos MQRFH2

Se IBM MQ converter propriedades de mensagem para sua representação MQRFH2, ele deverá incluir o MQRFH2 na mensagem. Ele inclui o MQRFH2 como um cabeçalho separado ou mescla com um cabeçalho existente.

A geração de novos cabeçalhos MQRFH2 por IBM MQ pode interromper cabeçalhos existentes em uma mensagem. Os aplicativos que analisam um buffer de mensagem para cabeçalhos devem estar cientes que o número e a posição de cabeçalhos em um buffer podem ser alterados em algumas circunstâncias IBM MQ tenta minimizar o impacto da inclusão de propriedades em uma mensagem mesclando propriedades de mensagem em um cabeçalho MQRFH2 existente, onde ele pode. Ele também tenta minimizar o impacto inserindo um MQRFH2 gerado em uma posição fixa relativa a outros cabeçalhos no buffer de mensagem.

Um cabeçalho MQRFH2 gerado é colocado após o MQMD de qualquer número de cabeçalhos MQXQH, MQRFH e MQDLH, independentemente da ordem em que estejam. O cabeçalho MQRFH2 gerado é colocado imediatamente antes do primeiro cabeçalho que não é MQMD, MQXQH, MQDLH ou MQRFH.

 Em sistemas z/OS, o cabeçalho MQRFH2 gerado é criado no CCSID do aplicativo. Isso é definido da seguinte forma:

- Para aplicativos LE em lote usando a interface DLL, o CCSID é o CODESET associado ao código do idioma atual no momento em que **MQCONN** é emitido (o valor padrão é 1047).
- Para aplicativos LE em lote ligados a um dos stubs MQ em lote, o CCSID é o CODESET associado ao código do idioma atual no momento da primeira chamada MQI emitida após **MQCONN** (o valor padrão é 1047).
- Para aplicativos não LE em lote executados em um encadeamento z/OS UNIX System Services (z/OS UNIX), o CCSID é o valor de THLICCSID no momento da primeira chamada MQI emitida após **MQCONN** (o valor padrão é 1047).
- Para outros aplicativos em lote, CCSID é o CCSID do gerenciador de fila.

Para aplicativos LE, o código de idioma pode ser alterado usando o serviço de chamada `setlocale()` / `CEESETL LE`. Para aplicativos não LE em execução em encadeamentos z/OS UNIX, o valor de THLICCSID pode ser alterado usando z/OS UNIX macro de mapeamento **BPXYTHLI**.

## Regras para mesclar MQRFH2 gerado

As regras a seguir se aplicam para mesclar um MQRFH2 gerado com um MQRFH2 existente. O cabeçalho MQRFH2 gerado será mesclado com um cabeçalho MQRFH2 existente, se:

1. O MQRFH2 existente está na mesma posição IBM MQ colocaria um MQRFH2 gerado ou anterior na cadeia de cabeçalho.
2. O CCSID das propriedades geradas é igual ao `NameValueCCSID` do MQRFH2 existente.

Caso contrário, o cabeçalho gerado será colocado separadamente no buffer, na posição descrita anteriormente.

## Regras para mesclar pastas em um MQRFH2 existente

Se as propriedades de mensagem forem mescladas em um MQRFH2 existente, o MQRFH2 existente será varrido para pastas que correspondam às propriedades de mensagem e as mesclará. Se uma pasta correspondente não existir, uma nova pasta será incluída no final das pastas existentes. Se uma pasta correspondente existir, ela será procurada. Todas as propriedades correspondentes são sobrescritas. Quaisquer novos são incluídos no final da pasta.

## Restrições da pasta MQRFH2

Visão geral de restrições de pasta em cabeçalhos MQRFH2

As restrições de MQRFH2 se aplicam às seguintes pastas:

- Nomes de elementos na pasta `<usr>` não devem começar com o prefixo `JMS`; Esses nomes de propriedades são reservados para uso pelo `JMS` e não são válidos para propriedades definidas pelo usuário.

Tal nome de elemento não faz com que a análise do MQRFH2 falhe, mas não é acessível para as APIs de propriedade de mensagens IBM MQ.

- Os nomes de elementos na pasta `<usr>` não devem ser, em nenhuma combinação de letras minúsculas ou maiúsculas, `NULL`, `TRUE`, `false`, `not` e `ou`, `between`, `like`, `IN`, `IS` e `ESCAPE`. Esses nomes correspondem a palavras-chaves SQL e tornam os seletores de análise mais difíceis, porque `<usr>` é a pasta padrão usada quando nenhuma pasta é especificada para uma propriedade específica em um seletor.

Tal nome de elemento não faz com que a análise do MQRFH2 falhe, mas não é acessível para as APIs de propriedade de mensagens IBM MQ.

- O modelo de conteúdo da pasta <usr> é o seguinte:
  - Qualquer nome XML válido pode ser usado como um nome de elemento, desde que não contenha dois pontos.
  - Apenas elementos simples, não pastas aninhadas, são permitidos
  - Todos os elementos usam o tipo padrão de sequência, a menos que seja modificado por um atributo dt="xxx"
  - Todos os elementos são opcionais, mas não devem ocorrer mais de uma vez em uma pastas.
- Os nomes de elementos em qualquer pasta considerada como contendo propriedades de mensagem não devem conter um ponto (.) (caractere Unicode U+002E), porque isso é usado em nomes de propriedade para indicar a hierarquia.

Tal nome de elemento não faz com que a análise do MQRFH2 falhe, mas não é acessível para as APIs de propriedade de mensagens IBM MQ .

Em geral, os cabeçalhos MQRFH2 que contêm dados de estilo XML válidos podem ser analisados pelo IBM MQ sem falha, embora determinados elementos do MQRFH2 não estejam acessíveis por meio das APIs de propriedades de mensagens IBM MQ .

## Conflitos de nome do elemento MQRFH2

Visão geral de conflitos dentro dos nomes de elementos MQRFH2 .

Apenas um valor pode ser conectado a uma propriedade de mensagem. Se uma tentativa de acessar uma propriedade leva a um conflito de valores, um é escolhido em preferência sobre outro.

A sintaxe IBM MQ para acessar elementos MQRFH2 permite a identificação exclusiva de um elemento, se uma pasta não contiver elementos com o mesmo nome. Se uma pasta contiver mais de um elemento com o mesmo nome, o valor da propriedade utilizada será o mais próximo do cabeçalho da mensagem..

Isso se aplica se duas ou mais pastas com o mesmo nome estiverem contidas em cabeçalhos MQRFH2 significativos diferentes na mesma mensagem.

Um conflito pode resultar quando a chamada MQGET for processada após uma propriedade do descritor de não mensagens ter sido configurada duas vezes: ambas por meio de uma chamada MQSETMP e diretamente no cabeçalho MQRFH2 bruto.

Se isso acontecer, a propriedade associada à mensagem por uma chamada API terá preferência sobre uma nos dados da mensagem, ou seja, aquela no cabeçalho MQRFH2 bruto. Se ocorrer um conflito, ele será considerado logicamente antes dos dados da mensagem..

## Mapeamento de nomes de propriedades para a pasta MQRFH2 e nomes de elementos

Visão geral das diferenças entre nomes de propriedade e nomes de elementos no cabeçalho MQRFH2 .

Ao usar qualquer uma das APIs definidas que finalmente geram cabeçalhos MQRFH2 , para especificar propriedades de mensagem (por exemplo, MQ JMS), o nome da propriedade não é necessariamente o nome do elemento na pasta MQRFH2 .

Portanto, um mapeamento ocorre a partir do nome da propriedade para o elemento MQRFH2 e, de forma inversa, levando em conta o nome da pasta que contém o elemento e o nome do elemento.. Alguns exemplos de IBM MQ classes for JMS já estão documentados em [Usando IBM MQ classes for Java](#)

Nome da Propriedade	Nome da pasta MQRFH2	Nome do elemento MQRFH2
JMSDestination	jms	Dst
JMSType	mcd	Type, Set, Fmt
xxx (definido pelo usuário, em que xxx não começa com JMS)	usr	xxx

Portanto, quando um aplicativo JMS acessa a propriedade JMSDestination , isso é mapeado para o elemento Dst na pasta <jms>

Ao especificar propriedades como elementos MQRFH2 , IBM MQ define seus elementos da seguinte forma:

Tabela 637. Nomes de Propriedades Mapeados para a Pasta MQRFH2 , Grupo e Nomes de Elementos			
Nome da Propriedade	Nome da pasta MQRFH2	Nome do grupo MQRFH2	Nome do elemento MQRFH2
<Property>	<usr>	n/a	<Property>
<folder>. <Property>	<folder>	n/a	<Property>
<folder>. <group>. <Property>	<folder>	<group>	<Property>

Por exemplo, quando um aplicativo IBM MQ tenta acessar a propriedade Property1 , isso é mapeado para o elemento Property1 na pasta <usr> A propriedade wmq . Property2 mapeia para a propriedade Property2 na pasta <wmq> .

Se o nome da propriedade contiver mais do que um . , o nome do elemento MQRFH2 usado será aquele após o final. e os grupos MQRFH2 são usados para formar uma hierarquia; grupos MQRFH2 aninhados são permitidos.

O cabeçalho JMS e as propriedades específicas do provedor que estão contidas em um MQRFH2 nas pastas <mcd>, <jms>e <mqext> são acessadas por um aplicativo IBM MQ usando os nomes abreviados definidos em [Usando IBM MQ classes for Java](#) .

JMS propriedades definidas pelo usuário são acessadas a partir da pasta <usr> . Um aplicativo IBM MQ pode usar a pasta <usr> para suas propriedades do aplicativo se for aceitável para que a propriedade apareça para aplicativos JMS como uma de suas propriedades definidas pelo usuário

Se não for aceitável, escolha outra pasta; a pasta <wmq\_usr> é fornecida como um local padrão para tais propriedades nãoJMS ..

Seus aplicativos podem especificar e usar qualquer pasta MQRFH2 com um uso bem definido, não documentado em “Propriedades especificadas como elementos MQRFH2” na página 963 se você observar o seguinte:

1. A pasta pode já estar em uso, ou pode ser usada no futuro, por outro aplicativo fornecendo acesso indefinido às propriedades contidas dentro dela; consulte [Nomes de propriedades](#) para a convenção de nomenclatura sugerida para nomes de propriedade
2. As propriedades não são acessíveis para versões anteriores do cliente IBM MQ classes for JMS ou XMS que podem acessar apenas a pasta <usr> para propriedades definidas pelo usuário..
3. A pasta deve ser marcada com o atributo content com o valor configurado como properties, por exemplo, content= 'properties '

“MQSETMP-Configurar propriedade de mensagem” na página 808 inclui automaticamente esse atributo, conforme necessário Esse atributo não deve ser incluído em nenhuma das pastas definidas pelo IBM, por exemplo, <jms> e <usr>.. Fazer isso faz com que a mensagem seja rejeitada pelo cliente IBM MQ classes for JMS antes IBM WebSphere MQ 7.0. com um MessageFormatException.

Como a pasta <usr> é o local padrão para propriedades da sintaxe <Property> , um aplicativo IBM MQ e um aplicativo JMS para acessar o mesmo valor da propriedade definido pelo usuário usando o mesmo nome.

## Nomes de pasta reservados

Há vários nomes de pastas reservadas. Não é possível usar nomes como seus prefixos de pastas; por exemplo, Root . Property1 não acessa uma propriedade válida porque Root está reservado A lista a seguir contém nomes de pasta reservados:

- Raiz
- Conteúdo

- Propriedades
- Meio ambiente
- LocalEnvironment
- DestinationList
- ExceptionList
- InputBody
- InputRoot
- InputProperties
- Ambiente Local da Entrada
- Lista de Destino da Entrada
- Lista de Exceções de Entrada
- OutputRoot
- Ambiente Local da Saída
- Lista de Destino da Saída
- Lista de Exceções da Saída

## Mapeando campos do descritor de propriedade para cabeçalhos MQRFH2

Quando uma propriedade é convertida em um elemento MQRFH2 , os atributos do elemento a seguir são usados para especificar os campos significativos do descritor de propriedade: Isso descreve como os campos MQPD são convertidos para os atributos do elemento MQRFH2 .

### Suporte

O campo do descritor de propriedade de Suporte é dividido em três atributos de elementos

- O atributo do elemento **sr** especifica valores na máscara de bits MQPD\_REJECT\_UNSUP\_MASK..
- O atributo de elemento **sa** especifica os valores na máscara de bits MQPD\_ACCEPT\_UNSUP\_MASK
- O atributo do elemento **sx** especifica valores na máscara de bits MQPD\_ACCEPT\_UNSUP\_IF\_XMIT\_MASK..

Esses atributos de elemento são válidos apenas na pasta < mq> e serão ignorados se forem configurados em elementos nas outras pastas que contêm propriedades

<i>Tabela 638. Campos MQPD mapeados para atributos do elemento MQRFH2</i>		
<b>Valor de suporte</b>	<b>atributo do elemento MQRFH2</b>	<b>Valor do atributo MQRFH2</b>
MQPD_SUPPORT_OPTIONAL	sa	opcional Esse é o valor-padrão.
MQPD_SUPPORT_REQUIRED	solicitação de serviço	requeridos
MQPD_SUPPORT_REQUIRED_IF_LOCAL	sx	locais

### Context

Use o atributo de elemento **context** para indicar o contexto de mensagem ao qual uma propriedade pertence Use apenas um valor. Esse atributo de elemento é válido em uma propriedade em qualquer pasta que contenha propriedades

Tabela 639. Valores de contexto mapeados para valores de atributo MQRFH2

Valor de Contexto	Valor do atributo MQRFH2
MQPD_NO_CONTEXT	Nenhum Esse é o valor-padrão.
MQPD_USER_CONTEXT	user

### CopyOptions

Use o atributo de elemento **copy** para indicar mensagens para as quais uma propriedade deve ser copiada. Mais de um valor é aceitável; separe vários valores com uma vírgula. Por exemplo, **copy='reply'** e **copy='publish,report'** são ambos válidos. Esse atributo de elemento é válido em uma propriedade em qualquer pasta que contenha propriedades.

**Nota:** Na definição de atributo, aspas simples ou aspas duplas são um uso válido, por exemplo, **copy='reply'** ou **copy="report"**.

Tabela 640. Valores de CopyOption mapeados para valores de atributo MQRFH2

Valor de CopyOption	Valor do atributo MQRFH2
MQPD_COPY_FORWARD	encaminhamento
MQPD_COPY_REPLY	resposta
MQPD_COPY_REPORT	relatório
MQPD_COPY_PUBLISH	publicar
MQPD_COPY_ALL	all Não especifique isso com nenhum outro valor.. Quando usado com outro valor, isso tem precedência sobre qualquer valor, exceto <b>none</b>
MQPD_COPY_DEFAULT	padrão Esse é o valor-padrão. Ele é equivalente a especificar os três valores MQCOPY_FORWARD, MQCOPY_REPORT e MQCOPY_PUBLISH Não especifique isso com nenhum outro valor..
MQPD_COPY_NONE	Nenhum Não especifique isso com nenhum outro valor.. Quando usado com outro valor, isso tem precedência.

### Restrições para a pasta < mq> MQRFH2

Quando uma mensagem é colocada em uma fila, é procurada uma pasta < mq> para que a mensagem possa ser processada de acordo com suas propriedades definidas pelo MQ. Para permitir a análise eficiente de propriedades definidas pelo MQ, as restrições a seguir se aplicam à pasta:

- Somente as propriedades na primeira pasta < mq> significativa na mensagem são atuadas pelo MQ; as propriedades em qualquer outra pasta < mq> na mensagem são ignoradas.
- Se a pasta estiver em UTF-8, apenas caracteres de byte único UTF-8 serão permitidos na pasta. Um caractere de multibyte na pasta pode causar falha na análise e a mensagem ser rejeitada.
- Não inclua grupos do MQRFH2 na pasta < mq>. A presença do caractere Unicode U+003C em um valor da propriedade fará com que a mensagem seja rejeitada.

- Não use sequências de escape na pasta Uma sequência de escape é tratada como o valor real do elemento.
- Apenas o caractere Unicode U+0020 é tratado como espaço em branco na pasta. Todos os outros caracteres são tratados como significativos e podem causar falha na análise da pasta e a mensagem ser rejeitada.

Se a análise da pasta < mq> falhar ou se a pasta não observar essas restrições, a mensagem será rejeitada com CompCode **MQCC\_FAILED** e Reason **MQRC\_RFH\_RESTRICTED\_FORMAT\_ERR**.

## Cabeçalhos MQRFH2 que não são válidos

No momento em que um MQPUT, MQPUT1 ou processos de chamada MQGET, uma análise parcial de quaisquer cabeçalhos MQRFH2 na mensagem pode ocorrer para verificar quais pastas estão incluídas e para determinar se as pastas contêm propriedades. Visão geral de cabeçalhos MQRFH2 que não são válidos.

Se a análise parcial da mensagem não puder ser concluída com êxito porque a estrutura não é válida, por exemplo, o campo StructLength é muito pequeno, então:

- A chamada MQPUT ou MQPUT1 falha com o código de razão MQRC\_RFH\_ERROR, se puder ser determinado que o aplicativo inclui alguma opção IBM WebSphere MQ 7 , para que os aplicativos existentes não falhem.
- A chamada MQGET é retornada com sucesso e o MQRFH2 contendo o erro é retornado no buffer fornecido.

Se a análise parcial falhar porque não é possível detectar se uma pasta específica contém propriedades ou não, por exemplo, a pasta inicia <<jms, portanto, a análise falha antes que o nome da pasta seja determinado, então:

- A chamada MQPUT ou MQPUT1 falha com o código de razão MQRC\_RFH\_FORMAT\_ERROR, se puder ser determinado que o aplicativo inclui alguma opção IBM WebSphere MQ 7 , para que os aplicativos existentes não falhem.
- A chamada MQGET é retornada com sucesso e o MQRFH2 contendo o erro é retornado no buffer fornecido.
- Enquanto internamente no gerenciador de filas, a mensagem não é rejeitada devido à pasta mal formatada, mas a pasta é sempre tratada como se nenhuma propriedade estivesse contida dentro dela.

Uma mensagem pode fluir pela rede do gerenciador de filas com uma pasta contendo esse erro de sintaxe, mas nunca sendo analisada e detectada, enquanto uma ou mais pastas na mensagem são:

- Válidos
- Analisado com sucesso
- Usado no processamento da mensagem

Portanto, a detecção não é garantida.

Se um de seus aplicativos usar “MQSETMP-Configurar propriedade de mensagem” na página 808 ou MQINQMP para acessar uma propriedade, e isso fizer com que uma pasta MQRFH2 seja totalmente analisada, detectando um erro de modo que a análise não possa ser concluída, isso será indicado por um código de retorno apropriado para a chamada API. Nenhuma propriedade na pasta é disponibilizada para o aplicativo

Se for feita uma tentativa de analisar completamente uma pasta MQRFH2 e o analisador localizar atributos de elemento não reconhecidos ou um tipo de dado não reconhecido, a análise continuará e será concluída com êxito sem nenhum aviso sendo emitido; isso não constitui um erro de análise.

## Conversão de página de códigos

Esta seção descreve nomes de conjunto de códigos e CCSIDs, idioma nacional, Conversão z/OS , Conversão IBM i , e suporte de conversões Unicode

Cada seção de idioma nacional lista as seguintes informações:

- Os CCSIDs nativos suportados
- As conversões de página de códigos não suportadas

Os termos a seguir são usados nas informações:

**AIX**

Indica IBM MQ for AIX..

**Linux**

Indica IBM MQ para Linux para Intel e IBM MQ para Linux para zSeries.

**IBM i OS/400**

Indica IBM MQ for IBM i..

**Windows**

Indica IBM MQ for Windows..

**z/OS**

Indica IBM MQ for z/OS..

O padrão para conversão de dados é para a conversão a ser executada no sistema de destino (recebimento).

Se o produto de origem suportar a conversão, um canal poderá ser configurado e os dados trocados configurando o atributo do canal CONVERT para YES na origem.

**Nota:**

1. A conversão para informações IBM MQ MQI client ocorre no servidor, portanto, o servidor deve suportar a conversão do CCSID do cliente para o CCSID do servidor.
2. A conversão pode incluir o suporte incluído pelo CSD/PTF na versão mais recente do IBM MQ Verifique o conteúdo do nível de serviço mais recente para ver se você precisa instalar um CSD/PTF para ativar essa conversão
3. O CCSID do gerenciador de filas IBM MQ deve ser Combinado ou SBCS.
4. Alguns CCSIDs, por exemplo 850 no AIX, que não são suportados pelo sistema operacional ainda podem ser usados pelo aplicativo e também podem ser configurados como o CCSID do gerenciador de filas IBM MQ . Isso é permitido apenas para o propósito de compatibilidade com versões anteriores e a conversão falhará se as tabelas de conversão relevantes não estiverem instaladas

Consulte Tabela 641 na página 972 para obter uma referência cruzada entre alguns dos números de CCSID e alguns nomes de conjunto de códigos da indústria.

**Referências relacionadas**

“Idiomas Nacionais” na página 973

Essas informações contêm os idiomas suportados pelo IBM MQ

**Nomes de conjunto de códigos e CCSIDs**

Os nomes do conjunto de códigos e os CCSIDs correspondentes para cada nome do conjunto de códigos

**z/OS** IBM MQ for z/OS fornece mais conversão do que é listado nas tabelas específicas do idioma. Para obter uma lista completa de conversões, consulte Tabela 674 na página 998

Tabela 641. Nomes de conjunto de códigos e CCSIDs	
Nomes do conjunto de códigos	CCSIDs
ISO 8859-1	819
ISO 8859-2	912



Tabela 641. Nomes de conjunto de códigos e CCSIDs (continuação)

<b>Nomes do conjunto de códigos</b>	<b>CCSIDs</b>
ISO 8859-3	913
ISO 8859-5	915
ISO 8859-6	1089
ISO 8859-7	813
ISO 8859-8	916
ISO 8859-9	920
ISO 8859-13	921
ISO 8859-15 (euro)	923
big5	950
eucJP	954 5050 33722
eucKR	970
eucTW	964
eucCN	1383
pck	943
GBK	1386
koi8-r	878

## Idiomas Nacionais

Essas informações contêm os idiomas suportados pelo IBM MQ






Os idiomas suportados pelo IBM MQ são:

- Inglês dos EUA-consulte o tópico [“Inglês americano”](#) na página 974
- Alemão-ver tópico [“Alemão”](#) na página 974
- Dinamarquês e norueguês-ver tópico [“Dinamarquês e norueguês”](#) na página 975
- Finlandês e Sueco-ver tópico [“Finlandês e Sueco”](#) na página 976
- Italiano-ver tópico [“Italiano”](#) na página 977
- Espanhol-ver tópico [“Espanhol”](#) na página 977
- Inglês do Reino Unido / Gaélico-ver tópico [“Inglês do Reino Unido /Gaelic”](#) na página 978
- Francês-ver tópico [“Francês”](#) na página 979
- Multilíngue-consulte o tópico [“Multilíngüe”](#) na página 979
- Português-ver tópico [“Português”](#) na página 980
- Islandês-ver tópico [“Islandês”](#) na página 981
- Línguas da Europa Oriental-ver tópico [“Línguas da Europa Oriental”](#) na página 981
- Cirílico-ver tópico [“Cirílico”](#) na página 983
- Estoniano-ver tópico [“Estoniano”](#) na página 983
- Letão e lituano-ver tópico [“Letão e lituano”](#) na página 985
- Ukranian-ver tópico [“Ucraniano”](#) na página 986
- Grego-ver tópico [“Grego”](#) na página 986
- Turco-ver tópico [“Turco”](#) na página 987

- Hebraico-ver tópico [“Hebraico”](#) na página 988
- Farsi-ver tópico [“Persa”](#) na página 990
- Urdu-ver tópico [“Urdu”](#) na página 990
- Tailandês-ver tópico [“Tailandês”](#) na página 990
- Lao-ver tópico [“Laosiano”](#) na página 991
- Vietnamita-ver tópico [“Vietnamês”](#) na página 991
- Japonês Latin SBCS-ver tópico [“SBCS em latim japonês”](#) na página 992
- Japonês Katakana SBCS-ver tópico [“Japonês Katakana SBCS”](#) na página 993
- Japonês Kanji / Latim Misto-ver tópico [“Japonês Kanji / Latim Misto”](#) na página 994
- Japonês Kanji / Katakana Mixed-ver tópico [“Kanji japonês / Katakana misturado”](#) na página 995
- Coreano-ver tópico [“Coreano”](#) na página 996
- Chinês simplificado-consulte o tópico [“Chinês simplificado”](#) na página 997
- Chinês tradicional-ver tópico [“Chinês Tradicional”](#) na página 997

### **Inglês americano**

Detalhes de CCSIDs e conversão CCSID para inglês dos EUA.

<i>Tabela 642. CCSIDs nativos para inglês dos EUA em plataformas suportadas</i>	
<b>Plataforma</b>	<b>CCSIDs nativos</b>
 IBM i  z/OS	37, 924, 1140
 AIX	819, 923, 5348
 Windows	437, 850, 1252, 5348, 858
 Linux	819, 923
Cliente Apple	1275

Todas as plataformas não clientes suportam a conversão entre seus CCSIDs nativos e os CCSIDs nativos das outras plataformas, com as exceções a seguir.

### **IBM i**



Página de códigos:

#### **37**

Não converte em páginas de códigos 923, 858

#### **924**

Não converte em páginas de códigos 437, 858, 1051, 1140, 1252, 1275, 5348






#### **1140**

Não converte em páginas de códigos 924, 1051, 1275

### **Alemão**

Detalhes de conversão de CCSIDs e CCSID para alemão.

Tabela 643. CCSIDs nativos para alemão em plataformas suportadas

Plataforma	CCSIDs nativos
 IBM i  z/OS	273, 924, 1141
 AIX	819, 923, 5348
 Windows	437, 850, 858, 1252, 5348
 Linux	819, 923
Cliente Apple	1275

Todas as plataformas não clientes suportam a conversão entre seus CCSIDs nativos e os CCSIDs nativos das outras plataformas, com as exceções a seguir.

## IBM i



Página de códigos:

### 273

Não converte em páginas de códigos 858, 923, 924, 1275

### 924

Não converte em páginas de códigos 273, 437, 858, 1051, 1141, 1252, 1275, 5348






### 1141

Não converte em páginas de códigos 924, 1051, 1275

## Dinamarquês e norueguês

Detalhes de conversão CCSIDs e CCSID para dinamarquês e norueguês.

Tabela 644. CCSIDs nativos para dinamarquês e norueguês em plataformas suportadas

Plataforma	CCSIDs nativos
 IBM i  z/OS	277, 924, 1142
 AIX	819, 923, 5348
 Windows	850, 858, 865, 1252, 5348
 Linux	819, 923
Cliente Apple	1275

Todas as plataformas não clientes suportam a conversão entre seus CCSIDs nativos e os CCSIDs nativos das outras plataformas, com as exceções a seguir.

## IBM i



Página de códigos:

### 277

Não converte em páginas de códigos 858, 923, 924, 1275

### 924

Não converte em páginas de códigos 277, 858, 865, 1051, 1142, 1252, 1275, 5348

### 1142

Não converte em páginas de códigos 924, 865, 1051, 1275

## AIX



Página de códigos:

### 819

Não converte para a página de códigos 865

## Windows



Página de códigos:

### 865

Não converte em páginas de códigos 1051, 1275

## Finlandês e Sueco

Detalhes de CCSIDs e conversão de CCSID para finlandês e sueco.

*Tabela 645. CCSIDs nativos para finlandês e sueco em plataformas suportadas*

Plataforma	CCSIDs nativos
IBM i	278, 924, 1143
z/OS	
AIX	819, 923, 5348
Windows	437, 850, 858, 865, 1252, 5348
Linux	819, 923
Cliente Apple	1275

Todas as plataformas não clientes suportam a conversão entre seus CCSIDs nativos e os CCSIDs nativos das outras plataformas, com as exceções a seguir.

## IBM i



Página de códigos:

### 278

Não converte em páginas de códigos 858, 923, 924, 1275

### 924

Não converte em páginas de códigos 278, 437, 858, 865, 1051, 1143, 1252, 1275, 5348

## 1143

Não converte em páginas de códigos 865, 924, 1051, 1275

## AIX



Página de códigos:

### 819

Não converte para a página de códigos 865

### 850

Não converte para a página de códigos 865

## Windows



Página de códigos:

### 865

Não converte em páginas de códigos 1051, 1275

## Italiano

Detalhes de conversão de CCSIDs e CCSID para italiano

<i>Tabela 646. CCSIDs nativos para italiano em plataformas suportadas</i>	
Plataforma	CCSIDs nativos
IBM i	280, 924, 1144
z/OS	
AIX	819, 923, 5348
Windows	437, 850, 858, 1252, 5348
Linux	819, 923
Cliente Apple	1275

Todas as plataformas não clientes suportam a conversão entre seus CCSIDs nativos e os CCSIDs nativos das outras plataformas, com as exceções a seguir.

## IBM i



Página de códigos:

### 280

Não converte em páginas de códigos 858, 923, 924, 1275

### 924

Não converte para páginas de códigos 280, 437, 858, 1051, 1144, 1252, 1275, 5348






### 1144

Não converte em páginas de códigos 924, 1051, 1275

## Espanhol

Detalhes de CCSIDs e conversão CCSID para espanhol.

Tabela 647. CCSIDs nativos para espanhol em plataformas suportadas

Plataforma	CCSIDs nativos
 IBM i  z/OS	284, 924, 1145
 AIX	819, 923, 5348
 Windows	437, 850, 858, 1252, 5348
 Linux	819, 923
Cliente Apple	1275

Todas as plataformas não clientes suportam a conversão entre seus CCSIDs nativos e os CCSIDs nativos das outras plataformas, com as exceções a seguir.

## IBM i



Página de códigos:

### 284

Não converte em páginas de códigos 858, 923, 924, 1275

### 924

Não converte em páginas de códigos 284, 437, 858, 1051, 1145, 1252, 1275, 5348






### 1145

Não converte em páginas de códigos 924, 1051, 1275

## Inglês do Reino Unido /Gaelic

Detalhes de conversão de CCSIDs e CCSID para inglês / gaélico do Reino Unido.

Tabela 648. CCSIDs nativos para inglês / gaélico do Reino Unido em plataformas suportadas

Plataforma	CCSIDs nativos
 IBM i  z/OS	285, 924, 1146
 AIX	819, 923, 5348
 Windows	437, 850, 858, 1252, 5348
 Linux	819, 923
Cliente Apple	1275

Todas as plataformas não clientes suportam a conversão entre seus CCSIDs nativos e os CCSIDs nativos das outras plataformas, com as exceções a seguir.

## IBM i



Página de códigos:

## 285

Não converte em páginas de códigos 858, 923, 924, 1275

## 924






Não converte em páginas de códigos 285, 437, 858, 1051, 1146, 1252, 1275, 5348

## 1146

Não converte em páginas de códigos 924, 1051, 1275

## Francês

Detalhes de CCSIDs e conversão CCSID para francês.

Plataforma	CCSIDs nativos
 IBM i  z/OS	297, 924, 1147
 AIX	819, 923, 5348
 Windows	437, 850, 858, 1252, 5348
 Linux	819, 923
Cliente Apple	1275

Todas as plataformas não clientes suportam a conversão entre seus CCSIDs nativos e os CCSIDs nativos das outras plataformas, com as exceções a seguir.

## IBM i



Página de códigos:

## 297

Não converte para páginas de códigos 858, 923, 924, 1275, 5348

## 924

Não converte para páginas de códigos 297, 437, 858, 1051, 1147, 1252, 1275, 5348

## 1147

Não converte em páginas de códigos 924, 1051, 1275

## Multilíngüe

Detalhes de CCSIDs e conversão CCSID para Multilíngüe.






Plataforma	CCSIDs nativos
 IBM i  z/OS	500, 924, 1148
 AIX	819, 923, 5348
 Windows	437, 850, 858, 1252, 5348

Tabela 650. CCSIDs nativos para conversão multilíngue em plataformas suportadas (continuação)

Plataforma	CCSIDs nativos
 Linux	819, 923
Cliente Apple	1275

Todas as plataformas não clientes suportam a conversão entre seus CCSIDs nativos e os CCSIDs nativos das outras plataformas, com as exceções a seguir.

## IBM i



Página de códigos:

### 500

Não converte em páginas de códigos 858, 923

### 924

Não converte para páginas de códigos 437, 858, 1051, 1148, 1252, 1275, 5348






### 1148

Não converte em páginas de códigos 924, 1051, 1275

## Português

Detalhes de CCSIDs e conversão CCSID para Português.

Tabela 651. CCSIDs nativos para português em plataformas suportadas

Plataforma	CCSIDs nativos
 IBM i	37, 500, 924, 1140
 z/OS	500, 924, 1140
 AIX	819, 923, 5348
 Windows	850, 858, 860, 1252, 5348
 Linux	819, 923
Cliente Apple	1275

Todas as plataformas não clientes suportam a conversão entre seus CCSIDs nativos e os CCSIDs nativos das outras plataformas, com as exceções a seguir.

## IBM i



Página de códigos:

### 37

Não converte em páginas de códigos 858, 923, 1275

### 500

Não converte em páginas de códigos 858, 923, 1275

### 924

Não converte para páginas de códigos 858, 860, 1051, 1140, 1252, 1275, 5348



## 1140

Não converte em páginas de códigos 860, 924, 1051, 1275

## Windows



Página de códigos:

### 860

Não converte em páginas de códigos 1051, 1275

## Islandês

Detalhes de CCSIDs e conversão CCSID para islandês.

*Tabela 652. CCSIDs nativos para islandês em plataformas suportadas*

Plataforma	CCSIDs nativos
IBM i z/OS	871, 924, 1149
AIX	819, 923, 5348
Windows	850, 858, 861, 1252, 5348
Linux	819, 923
Cliente Apple	1275

Todas as plataformas não clientes suportam a conversão entre seus CCSIDs nativos e os CCSIDs nativos das outras plataformas, com as exceções a seguir.

## IBM i



Página de códigos:

### 871

Não converte para páginas de códigos 858, 923, 924, 1275, 5348

### 924

Não converte em páginas de códigos 858, 861, 871, 1051, 1149, 1252, 1275, 5348

### 1149

Não converte em páginas de códigos 924, 1051, 1275

## Windows



Página de códigos:






### 861

Não converte em páginas de códigos 1051, 1275

## Línguas da Europa Oriental

Detalhes de CCSIDs e conversão de CCSID para Idiomas do Leste Europeu. As línguas típicas que usam esses CCSIDs incluem albanês, croata, checo, húngaro, polonês, romeno, sérvio, eslovaco e esloveno.

Tabela 653. CCSIDs nativos para idiomas do Leste Europeu em plataformas suportadas

Plataforma	CCSIDs nativos
 IBM i  z/OS	870, 1153
 Windows	852, 1250, 5346, 9044
 AIX  Linux	912
Cliente Apple do Leste Europeu	1282
Cliente Apple romeno	1285
Cliente Apple da Croácia	1284

Todas as plataformas não clientes suportam a conversão entre seus CCSIDs nativos e os CCSIDs nativos das outras plataformas, com as exceções a seguir.

## z/OS



Página de códigos:

### 870

Não converte para páginas de códigos 1284, 1285

### 1153

Não converte em páginas de códigos 1250, 1284, 1285

## IBM i



Página de códigos:

### 870

Não converte em páginas de códigos 1284, 1285, 5346, 9044

### 1153

Não converte para páginas de códigos 1282, 1284, 1285, 5346, 9044

## , Linux



Página de códigos:

### 912

Não converte para páginas de códigos 1284, 1285

## Windows



Página de códigos:

### 852

Não converte para páginas de códigos 1284, 1285

**1250**

Não converte para páginas de códigos 1284, 1285






**9044**

Não converte para páginas de códigos 912, 1282, 1284, 1285

**Cirílico**

Detalhes de CCSIDs e conversão de CCSID para cirílico. As línguas típicas que usam esses CCSIDs incluem Belarussian, búlgaro, macedônio, russo e sérvio.

*Tabela 654. CCSIDs nativos para cirílico em plataformas suportadas*

Plataforma	CCSIDs nativos
 z/OS	1025
 IBM i	880, 1025
 Windows	855, 866, 1131, 1251, 5347
 AIX	915
 Linux	
Cliente Apple	1283

Todas as plataformas não clientes suportam a conversão entre seus CCSIDs nativos e os CCSIDs nativos das outras plataformas, com as exceções a seguir.

**IBM i**

Página de códigos:

**880**

Não converte em páginas de códigos 855, 866, 878, 1131, 5347

**1025**

Não converte em páginas de códigos 878, 5347

**Windows**

Página de códigos:

**855**

Não converte para a página de códigos 1131

**866**

Não converte para a página de códigos 1131






**1131**

Não converte em páginas de códigos 855, 866, 880, 1283

**Estoniano**

Detalhes de conversão de CCSIDs e CCSID para estoniano.

Tabela 655. CCSIDs nativos para estoniano em plataformas suportadas

Plataforma	CCSIDs nativos
 IBM i  z/OS	1122, 1157
 Windows	902, 922, 1257, 5353, 9449
 AIX  Linux	902, 922

Todas as plataformas suportam conversão entre seus CCSIDs nativos e os CCSIDs nativos de outras plataformas, com as exceções a seguir.

## z/OS



Página de códigos:

### 1122

Não converte em páginas de códigos 902, 1157, 9449

### 1157

Não converte em páginas de códigos 922, 1122, 1257, 9449

## IBM i



Página de códigos:

### 1122

Não converte em páginas de códigos 902, 5353, 9449

### 1157

Não converte para páginas de códigos 922, 5353, 9449

## Linux



Página de códigos:

### 902

Não converte em páginas de códigos 922, 1122, 9449

### 922

Não converte em páginas de códigos 902, 1157, 9449

## Windows



Página de códigos:

### 5353

Não converte para a página de códigos 9449

### 9449






Não converte para as páginas de códigos 902, 922, 1122, 1157, 1257, 5353

## 902

Não converte em páginas de códigos 922, 1122, 9449

## Letão e lituano

Detalhes de conversão de CCSIDs e CCSID para letão e lituano.

<i>Tabela 656. CCSIDs nativos para letão e lituano em plataformas suportadas</i>	
Plataforma	CCSIDs nativos
 IBM i  z/OS	1112, 1156
 Windows	901, 921, 1257, 5353, 9449
 AIX  Linux	901, 921

Todas as plataformas suportam conversão entre seus CCSIDs nativos e os CCSIDs nativos de outras plataformas, com as exceções a seguir.

## z/OS



Página de códigos:

### 1112

Não converte em páginas de códigos 901, 1156, 9449

### 1156

Não converte em páginas de códigos 901, 1156, 9449

## IBM i



Página de códigos:

### 1112

Não converte para a página de códigos 5353

### 1153

Não converte em páginas de códigos 921, 5353, 9449

## Linux



Página de códigos:

### 902

Não converte em páginas de códigos 921, 1112, 1257, 9449

### 921

Não converte em páginas de códigos 901, 1156, 9449

## Windows



Página de códigos:

## 901

Não converte em páginas de códigos 921, 1112, 1257, 9449

## 5355






Não converte para a página de códigos 9449

## 9449

Não converte em páginas de códigos 901, 921, 1112, 1156, 1257

## Ucraniano

Detalhes de conversão de CCSIDs e CCSID para ucraniano.

<i>Tabela 657. CCSIDs nativos para Ukranian em plataformas suportadas</i>	
Plataforma	CCSIDs nativos
 IBM i	1123
 z/OS	
 Windows	1124, 1125, 1251, 5347
 AIX	1124
 Linux	

Todas as plataformas suportam conversão entre seus CCSIDs nativos e os CCSIDs nativos de outras plataformas, com as exceções a seguir.

## IBM i



Página de códigos:

### 1123

Não converte para a página de códigos 5347

## Windows



Página de códigos:

### 1125

Não converte para a página de códigos 1123

## Grego

Detalhes de conversão de CCSIDs e CCSID para grego.






<i>Tabela 658. CCSIDs nativos para grego em plataformas suportadas</i>	
Plataforma	CCSIDs nativos
 IBM i	875
 z/OS	
 Windows	869, 1253, 5349

Tabela 658. CCSIDs nativos para grego em plataformas suportadas (continuação)

Plataforma	CCSIDs nativos
 AIX  Linux NCR	813
Cliente Apple	12 80
Cliente DOS	737

Todas as plataformas não clientes suportam conversão entre seus CCSIDs nativos, os CCSIDs nativos das outras plataformas com as exceções a seguir.

## IBM i



Página de códigos:

### 875

Não converte para a página de códigos 5349

## Windows



Página de códigos:

### 1253

Não converte para a página de códigos 737






### 5349

Não converte para a página de códigos 737

## Turco

Detalhes de conversão de CCSIDs e CCSID para turco.

Tabela 659. CCSIDs nativos para turco em plataformas suportadas

Plataforma	CCSIDs nativos
 IBM i  z/OS	1026
 Windows	857, 1254, 5350
 AIX  Linux	920
Cliente Apple	1281

Todas as plataformas não clientes suportam a conversão entre seus CCSIDs nativos e os CCSIDs nativos das outras plataformas, com as exceções a seguir.

## IBM i



Página de códigos:

### 1026

Não converte para a página de códigos 5350

## Hebraico

Detalhes de CCSIDs e conversão CCSID para hebraico..

Plataforma	CCSIDs nativos
z/OS	424, 803, 4899, 12712
IBM i	424
AIX	916, 9048
Windows	1255, 5351
Linux	916

Todas as plataformas suportam conversão entre seus CCSIDs nativos e os CCSIDs nativos de outras plataformas, com as exceções a seguir.

## z/OS



Página de códigos:

### 424

Não converte para páginas de códigos 867, 4899, 9048, 12712

### 803

Não converte em páginas de códigos 867, 4899, 5351, 9048, 12712

### 4899

Não converte para páginas de códigos 424, 803, 856, 862, 916, 1255

### 12712

Não converte para páginas de códigos 424, 803, 856, 916, 1255

## IBM i



Página de códigos:

### 424

Não converte para páginas de códigos 803, 867, 4899, 5351, 9048, 12712

A página de códigos 424 também converte de e para CCSID 4952, que é uma variante de 856.

## AIX



Página de códigos:



## 916

Não converte para páginas de códigos 867, 4899, 9048, 12712

## 9048

Não converte para páginas de códigos 424, 803, 856, 862, 916, 1255

## Windows



Página de códigos:

### 1255

Não converte para páginas de códigos 867, 4899, 9048, 12712

### 5351

Não converte para a página de códigos 803

## árabe

Detalhes de conversão de CCSIDs e CCSID para árabe

Plataforma	CCSIDs nativos
IBM i	420
z/OS	
AIX	1046, 1089
	1089 (ver nota)
Windows	720, 864, 1256, 5352
Linux	1089

Todas as plataformas suportam conversão entre seus CCSIDs nativos e os CCSIDs nativos de outras plataformas, com as exceções a seguir.

## IBM i



Página de códigos:

### 420

Não converte para a página de códigos 5352

## Linux, Tru64



Página de códigos:

### 1089

Não converte para a página de códigos 720

## Windows



Página de códigos:

## 720






Não converte em páginas de códigos 1089, 5352

## 5352

Não converte para a página de códigos 720

## Persa

Detalhes de CCSIDs e conversão de CCSID para Farsi.






Plataforma	CCSIDs nativos
 IBM i  z/OS	1097
 AIX  Linux  Windows	1098 (ver nota)

**Nota:** O CCSID nativo para essas plataformas não foi padronizado e pode ser alterado.

Todas as plataformas suportam conversão entre seus CCSIDs nativos e os CCSIDs nativos de outras plataformas.

## Urdu

Detalhes de CCSIDs e conversão CCSID para Urdu.

Plataforma	CCSIDs nativos
 IBM i  z/OS	918
 Windows	868
 AIX  Linux	1006

Todas as plataformas suportam conversão entre seus CCSIDs nativos e os CCSIDs nativos de outras plataformas, com as exceções a seguir.

## IBM i



Página de códigos:






## 918

Não converte para a página de códigos 1006

## Tailandês

Detalhes de conversão de CCSIDs e CCSID para Thai.

Tabela 664. CCSIDs nativos para Thai em plataformas suportadas

Plataforma	CCSIDs nativos
 IBM i  z/OS	838
 AIX  Linux  Windows	874 (ver nota)






**Nota:** O CCSID nativo para essas plataformas não foi padronizado e pode ser alterado.

Todas as plataformas suportam conversão entre seus CCSIDs nativos e os CCSIDs nativos de outras plataformas.

### Laosiano

Detalhes de conversão de CCSIDs e CCSID para Lao.

Tabela 665. CCSIDs nativos para Lao em plataformas suportadas






Plataforma	CCSIDs nativos
 IBM i  z/OS	1132
 AIX  Linux  Windows	1133

Todas as plataformas suportam conversão entre seus CCSIDs nativos e os CCSIDs nativos de outras plataformas.

### Vietnamês

Detalhes de CCSIDs e conversão de CCSID para vietnamita.

Tabela 666. CCSIDs nativos para vietnamita em plataformas suportadas

Plataforma	CCSIDs nativos
 IBM i  z/OS	1130
 Windows	1258, 5354
 AIX  Linux	1129

Todas as plataformas suportam conversão entre seus CCSIDs nativos e os CCSIDs nativos de outras plataformas, com as exceções a seguir.

## IBM i



Página de códigos:

### 1130

Não converte em páginas de códigos 1129, 5354

## SBCS em latim japonês

Detalhes de conversão de CCSIDs e CCSID para SBCS latino japonês.

Plataforma	CCSIDs nativos
IBM i z/OS	1027
AIX	932, 5050, 33722 (ver Nota 1)
Windows	932, 943 (ver nota 2)
Linux	943, 5050

### Nota:

- 5050 e 33722 são CCSIDs relacionados à página de códigos de base 954 em AIX O CCSID reportado pelo sistema operacional é 33722.
- Windows NT usa a página de códigos 932, mas isso é melhor representado pelo CCSID de 943. Entretanto, nem todas as plataformas do IBM MQ suportam esse CCSID.  
No IBM MQ for Windows , o CCSID 932 é usado para representar a página de códigos 932, mas uma mudança no arquivo `./conv/table/ccsid.tbl` pode ser feita que altera o CCSID usado para 943.

Todas as plataformas suportam conversão entre seus CCSIDs nativos e os CCSIDs nativos de outras plataformas, com as exceções a seguir.

## z/OS



Página de códigos:

### 1027

Não converte para páginas de códigos 932, 942, 943, 954, 5050, 33722

## IBM i



Página de códigos:

### 1027

Não converte para a página de códigos 932

## AIX



Página de códigos:

**932**

Não converte para a página de códigos 1027

**5050**

Não converte para a página de códigos 1027

**33722**

Não converte para a página de códigos 1027

**Linux**

Página de códigos:

**943**

Não converte para a página de códigos 1027

**5050**

Não converte para a página de códigos 1027

**Japonês Katakana SBCS**

Detalhes de CCSIDs e conversão de CCSID para SBCS japonês Katakana.

*Tabela 668. CCSIDs nativos para japonês Katakana SBCS em plataformas suportadas*

Plataforma	CCSIDs nativos
IBM i z/OS	290
AIX	932, 5050, 33722 (ver Nota 1)
Windows	932, 943 (ver nota 2)
Linux	943, 5050

**Nota:**

- 5050 e 33722 são CCSIDs relacionados à página de códigos de base 954 em AIX O CCSID reportado pelo sistema operacional é 33722.
- Windows NT usa a página de códigos 932, mas isso é melhor representado pelo CCSID de 943. Entretanto, nem todas as plataformas do IBM MQ suportam esse CCSID.  
 No IBM MQ for Windows , o CCSID 932 é usado para representar a página de códigos 932, mas uma mudança no arquivo `./conv/table/ccsid.tbl` pode ser feita que altera o CCSID usado para 943.
- Além das conversões anteriores, o IBM MQ suporta a conversão de CCSID 897 para CCSIDs 37, 273, 277, 278, 280, 284, 285, 290, 297, 437, 500, 819, 850, 1027 e 1252 nas plataformas a seguir:
  - AIX
  - Linux

Todas as plataformas suportam conversão entre seus CCSIDs nativos e os CCSIDs nativos de outras plataformas, com as exceções a seguir.

**z/OS**

Página de códigos:

## 290

Não converte para páginas de códigos 932, 943, 954, 5050, 33722

## IBM i



Página de códigos:

## 290

Não converte para a página de códigos 932

## AIX



Página de códigos:

## 932

Não converte em páginas de códigos 290, 897

## 5050

Não converte em páginas de códigos 290, 897

## 33722

Não converte em páginas de códigos 290, 897

## Linux



Página de códigos:

## 943

Não converte em páginas de códigos 290, 897

## 5050

Não converte em páginas de códigos 290, 897

## Japonês Kanji / Latim Misto

Detalhes de CCSIDs e conversão de CCSID para Kanji japonês / latim misturado.

Plataforma	CCSIDs nativos
IBM i z/OS	1399, 5035 (ver Nota 1)
AIX	932, 5050, 33722 (ver Nota 2)
Windows	932, 943 (ver nota 4)
Linux	943, 5050

### Nota:

- 5035 é um CCSID relacionado à página de códigos 939
- 5050 e 33722 são CCSIDs relacionados à página de códigos de base 954 em AIX O CCSID reportado pelo sistema operacional é 33722.

3. **Windows** Windows NT usa a página de códigos 932, mas isso é melhor representado pelo CCSID de 943. Entretanto, nem todas as plataformas do IBM MQ suportam esse CCSID.

No IBM MQ for Windows, o CCSID 932 é usado para representar a página de códigos 932, mas uma mudança no arquivo `./conv/table/ccsid.tbl` pode ser feita que altera o CCSID usado para 943.

Todas as plataformas suportam conversão entre seus CCSIDs nativos e os CCSIDs nativos de outras plataformas, com as exceções a seguir.

## z/OS



Página de códigos:

### 1399

Não converte para páginas de códigos 954, 5035, 5050, 33722

### 5035

Não converte em páginas de códigos 954, 1399, 5050, 33722

## IBM i



Página de códigos:

### 1399

Não converte para a página de códigos 5039

### 5035

Não converte para a página de códigos 5039

## Kanji japonês / Katakana misturado

Detalhes de CCSIDs e conversão de CCSID para Kanji japonês / Katakana Mixed.

Plataforma	CCSIDs nativos
z/OS	1390, 5026 (ver Nota “1” na página 995)
IBM i	5026 (ver Nota “1” na página 995)
AIX	932, 5050, 33722 (ver Nota “2” na página 996)
Windows	932, 943 (ver Nota “3” na página 996)
Linux	943, 5050

### Nota:

1. O modo de byte único de CCSIDs 1390 e 5026 em EBCDIC contém caracteres minúsculos em locais diferentes para o layout típico ou invariante para latim básico. Portanto, deve-se tomar cuidado para assegurar que os dados não sejam perdidos quando os dados da mensagem forem convertidos em outros CCSIDs Além disso, o uso desses CCSIDs como um CCSID padrão do gerenciador de filas pode causar problemas ao se comunicar com outros gerenciadores de filas. Por exemplo, os nomes de canal usando caracteres minúsculos podem não ser interpretados corretamente no sistema remoto. 5026 é um CCSID relacionado à página de códigos 930. CCSID 5026 é o CCSID relatado no IBM i quando o recurso japonês Katakana (DBCS) é selecionado.

2. **AIX** 5050 e 33722 são CCSIDs relacionados à página de códigos de base 954 em AIX O CCSID reportado pelo sistema operacional é 33722.
3. **Windows** Windows NT usa a página de códigos 932, mas isso é melhor representado pelo CCSID de 943. Entretanto, nem todas as plataformas do IBM MQ suportam esse CCSID.

No IBM MQ for Windows, o CCSID 932 é usado para representar a página de códigos 932, mas uma mudança no arquivo `./conv/table/ccsid.tbl` pode ser feita e o CCSID usado para 943.

Todas as plataformas suportam conversão entre seus CCSIDs nativos e os CCSIDs nativos de outras plataformas, com as exceções a seguir.

## z/OS



Página de códigos:

### 1390

Não converte em páginas de códigos 954, 5026, 5050, 33722

Não aceita caracteres minúsculos

### 5026

Não converte em páginas de códigos 954, 1390, 5050, 33722

## IBM i



Página de códigos:

### 5026

Não converte em páginas de códigos 1390, 5039

## Coreano

Detalhes de conversão de CCSIDs e CCSID para coreano.

Plataforma	CCSIDs nativos
IBM i z/OS	933, 1364
AIX Linux	970
Windows	949, 1363

Todas as plataformas suportam conversão entre seus CCSIDs nativos e os CCSIDs nativos de outras plataformas, com as exceções a seguir.

## z/OS



Página de códigos:

### 933

Não converte para a página de códigos 970








## 1364


Não converte para a página de códigos 970

### Chinês simplificado

Detalhes de CCSIDs e conversão CCSID para chinês simplificado.




Plataforma	CCSIDs nativos
 z/OS	935, 1388
 IBM i	935, 1388
 AIX	1383, 1386
 Windows	1381, 1386 (ver nota 2)
 Linux	1383

#### Nota:


1.  O Windows usa a página de códigos 936, mas isso é melhor representado pelo CCSID de 1386. Entretanto, nem todas as plataformas do IBM MQ suportam esse CCSID.

No IBM MQ for Windows CCSID 1381 é usado para representar a página de códigos 936, mas uma mudança no arquivo `./conv/table/ccsid.tbl` pode ser feita que altera o CCSID usado para 1386.

2. IBM MQ suporta o padrão chinês GB18030 .

   No z/OS, Windows e Linux, o suporte de conversão é fornecido entre Unicode (UTF-8 e UTF-16) e CCSID 1388 (EBCDIC com extensões GB18030 ), Unicode (UTF-8 e UTF-16) e CCSID 5488 (GB18030) e entre CCSID 1388 e CCSID 5488.

**Nota:** O CCSID deve ser configurado como 5488 para usar GB18030 caracteres. No entanto, não é possível configurar o CCSID em um gerenciador de filas criado com o IBM MQ Explorer ou o IBM MQ Console. Em vez disso, você deve criar o gerenciador de filas usando a CLI com um CCSID de 5488 ou usar a linha de comandos da CLI para mudar o CCSID depois de criar o gerenciador de filas.

 No IBM i, o suporte é fornecido pelo sistema operacional para conversão entre Unicode (UTF-8 e UTF-16) e CCSID 1388 (EBCDIC com extensões GB18030 ).

Todas as plataformas suportam conversão entre seus CCSIDs nativos e os CCSIDs nativos de outras plataformas, com as exceções a seguir.

### z/OS



Página de códigos:

#### 935

Não converte para a página de códigos 1383






#### 1388

Não converte para a página de códigos 1383

### Chinês Tradicional

Detalhes de CCSIDs e conversão CCSID para chinês tradicional.

Tabela 673. CCSIDs nativos para chinês tradicional em plataformas suportadas

Plataforma	CCSIDs nativos
 IBM i  z/OS	937
 Windows	950
 AIX  Linux	950, 964

Todas as plataformas suportam conversão entre seus CCSIDs nativos e os CCSIDs nativos de outras plataformas, com as exceções a seguir.

### z/OS



Página de códigos:

#### 937

Não converte para a página de códigos 964

#### 1388

Não converte para a página de códigos 1383

### Linux



Página de códigos:

#### 964

Não converte para a página de códigos 938

### Suporte de conversão z/OS

Uma lista de conversões CCSID suportadas.

Tabela 674. IBM MQ for z/OS Suporte à conversão de CCSID

CCSID	Converte para e de CCSIDS
37	256, 273, 275, 277-278, 280, 284-285, 290, 297, 367, 420, 423-424, 437, 500, 720, 737, 775, 813, 819, 833, 836, 838, 850, 852, 855, 857-858, 860-866, 869-871, 874-875, 880, 897, 903-905, 912, 914-916, 920-924, 1009, 1025-1027, 1040-1043, 1047, 1051, 1088, 1097, 1100, 1112, 1114-1115, 1122, 1124, 1126, 1130-1132, 1137, 1140-1149, 1200, 1208, 1250-1255, 1257-1258, 1275, 1280-1281, 1283, 4386, 4909, 4929, 4932, 4934, 4946, 4948, 4951, 4953, 4960, 4970-4971, 5012, 5123, 5210-5211, 5346, 5348, 8229, 8482, 8612, 9025, 9030, 9044, 9049, 9056, 9061, 9066, 13121, 13488, 16804, 17248, 17584, 25473, 25479, 25480, 25617, 25619, 25664, 28709

Tabela 674. IBM MQ for z/OS Suporte à conversão de CCSID (continuação)

CCSID	Converte para e de CCSIDS
256	37, 273, 277-278, 280, 284-285, 290, 297, 367, 420, 423-424, 437, 500, 737, 775, 819, 833, 836, 838, 850, 852, 857, 860-866, 869-871, 875, 880, 905, 1025-1027, 1112, 1122, 1200, 1208, 1251-1252, 1275, 4386, 4929, 4932, 4934, 4946, 4948, 4953, 4960, 4971, 5123, 8229, 8482, 8612, 9025, 9030, 9044, 9049, 9056, 9061, 13121, 13488, 16804, 17248, 17584, 28709
259	437, 808, 850-852, 855-858, 860-865, 867, 869, 872, 874, 899, 901-902, 915, 1098, 1161-1162, 1200, 1208, 1250-1258, 4946, 4948, 4951-4953, 4960, 4970, 5346, 5348, 9044, 9049, 9056, 9061, 9066, 13488, 17248, 17584
273	37, 256, 277-278, 280, 284-285, 290, 297, 367, 423, 437, 500, 737, 775, 813, 819, 833, 836, 838, 850, 852, 855-858, 860-865, 869-871, 874-875, 880, 897, 903, 912, 916, 920, 923-924, 1009, 1025-1027, 1040-1043, 1047, 1051, 1088, 1100, 1112, 1122, 1140-1149, 1200, 1208, 1250, 1252, 1275, 4386, 4909, 4929, 4932, 4934, 4946, 4948, 4951-4953, 4960, 4970-4971, 5012, 5123, 5346, 5348, 8229, 8482, 9025, 9030, 9044, 9049, 9056, 9061, 9066, 13121, 13488, 17248, 17584, 25473, 25479, 25617, 25619, 25664, 28709
274	500, 1047
275	37, 437, 500, 819, 850, 1047, 1200, 1208, 1252, 4946, 5348, 8229, 13488, 17584, 28709
277	37, 256, 273, 278, 280, 284-285, 290, 297, 367, 423, 437, 500, 737, 775, 813, 819, 833, 836, 838, 850, 852, 855, 857-858, 860-865, 869-871, 874-875, 880, 897, 903, 912, 916, 920, 923-924, 1009, 1025-1027, 1040-1043, 1047, 1051, 1088, 1100, 1112, 1122, 1140-1149, 1200, 1208, 1252, 1275, 4386, 4909, 4929, 4932, 4934, 4946, 4948, 4951, 4953, 4960, 4970-4971, 5012, 5123, 5348, 8229, 8482, 9025, 9030, 9044, 9049, 9056, 9061, 9066, 13121, 13488, 17248, 17584, 25473, 25479, 25617, 25619, 25664, 28709
278	37, 256, 273, 277, 280, 284-285, 290, 297, 367, 423, 437, 500, 737, 775, 813, 819, 833, 836, 838, 850, 852, 855, 857-858, 860-865, 869-871, 874-875, 880, 897, 903, 912, 916, 920, 923-924, 1009, 1025-1027, 1040-1043, 1047, 1051, 1088, 1100, 1112, 1122, 1140-1149, 1200, 1208, 1252, 1275, 4386, 4909, 4929, 4932, 4934, 4946, 4948, 4951, 4953, 4960, 4970-4971, 5012, 5123, 5348, 8229, 8482, 9025, 9030, 9044, 9049, 9056, 9061, 9066, 13121, 13488, 17248, 17584, 25473, 25479, 25617, 25619, 25664, 28709
280	37, 256, 273, 277-278, 284-285, 290, 297, 367, 423, 437, 500, 737, 775, 813, 819, 833, 836, 838, 850, 852, 855, 857-858, 860-865, 869-871, 874-875, 880, 897, 903, 912, 916, 920, 923-924, 1009, 1025-1027, 1040-1043, 1047, 1051, 1088, 1100, 1112, 1122, 1140-1149, 1200, 1208, 1252, 1275, 4386, 4909, 4929, 4932, 4934, 4946, 4948, 4951, 4953, 4960, 4970-4971, 5012, 5123, 5348, 8229, 8482, 9025, 9030, 9044, 9049, 9056, 9061, 9066, 13121, 13488, 17248, 17584, 25473, 25479, 25617, 25619, 25664, 28709
281	1047
282	500, 1047, 1200, 1208, 13488, 17584

Tabela 674. IBM MQ for z/OS Suporte à conversão de CCSID (continuação)

CCSID	Converte para e de CCSIDS
284	37, 256, 273, 277-278, 280, 285, 290, 297, 367, 423, 437, 500, 737, 775, 813, 819, 833, 836, 838, 850, 852, 855, 857-858, 860-865, 869-871, 874-875, 880, 897, 903, 912, 916, 920, 923-924, 1009, 1025-1027, 1040-1043, 1047, 1051, 1088, 1100, 1112, 1122, 1140-1149, 1200, 1208, 1252, 1275, 4386, 4909, 4929, 4932, 4934, 4946, 4948, 4951, 4953, 4960, 4970-4971, 5012, 5123, 5348, 8229, 8482, 9025, 9030, 9044, 9049, 9056, 9061, 9066, 13121, 13488, 17248, 17584, 25473, 25479, 25617, 25619, 25664, 28709
285	37, 256, 273, 277-278, 280, 284, 290, 297, 423, 437, 500, 737, 775, 813, 819, 833, 836, 838, 850, 852, 855, 857-858, 860-865, 869-871, 874-875, 880, 897, 903, 912, 916, 920, 923-924, 1025-1027, 1040-1043, 1047, 1051, 1088, 1100, 1112, 1122, 1140-1149, 1200, 1208, 1252, 1275, 4386, 4909, 4929, 4932, 4934, 4946, 4948, 4951, 4953, 4960, 4970-4971, 5012, 5123, 5348, 8229, 8482, 9025, 9030, 9044, 9049, 9056, 9061, 9066, 13121, 13488, 17248, 17584, 25473, 25479, 25617, 25619, 25664, 28709
290	37, 256, 273, 277-278, 280, 284-285, 297, 367, 437, 500, 737, 775, 819, 833, 836, 850, 852, 855, 857, 860-865, 870-871, 895-897, 1009, 1025-1027, 1040-1043, 1047, 1088, 1112, 1122, 1139, 1200, 1208, 1252, 4386, 4929, 4932, 4946, 4948, 4951, 4953, 4960, 4992, 5123, 8229, 8482, 9025, 9044, 9049, 9056, 13121, 13488, 17248, 17584, 25473, 25617, 25619, 25664, 28709
293	1200, 1208, 13488, 17584
297	37, 256, 273, 277-278, 280, 284-285, 290, 367, 423, 437, 500, 737, 775, 813, 819, 833, 836, 838, 850, 852, 855, 857-858, 860-865, 869-871, 874-875, 880, 897, 903, 912, 916, 920, 923-924, 1009, 1025-1027, 1040-1043, 1047, 1051, 1088, 1100, 1112, 1122, 1140-1149, 1200, 1208, 1252, 1275, 4386, 4909, 4929, 4932, 4934, 4946, 4948, 4951, 4953, 4960, 4970-4971, 5012, 5123, 5348, 8229, 8482, 9025, 9030, 9044, 9049, 9056, 9061, 9066, 13121, 13488, 17248, 17584, 25473, 25479, 25617, 25619, 25664, 28709
300	301, 941, 1200, 1208, 1351, 4396, 8492, 13488, 16684, 17584
301	300, 941, 1200, 1208, 1351, 4396, 8492, 13488, 16684, 17584
367	37, 256, 273, 277-278, 280, 284, 290, 297, 500, 819, 833, 836, 850, 871, 875, 1009, 1026-1027, 1041, 1088, 1115, 1126, 1200, 1208, 4386, 4929, 4932, 4946, 4971, 5123, 5211, 8229, 8482, 9025, 13121, 13488, 17584, 25617, 25664, 28709
420	37, 256, 424, 437, 500, 720, 737, 775, 819, 850, 852, 857, 860-865, 1008, 1046, 1089, 1098, 1112, 1122, 1127, 1200, 1208, 1252, 1256, 4946, 4948, 4953, 4960, 5104, 5142, 5352, 8229, 8612, 9044, 9049, 9056, 9238, 13488, 16804, 17248, 17584, 28709
423	37, 256, 273, 277-278, 280, 284-285, 297, 437, 500, 737, 775, 813, 819, 838, 850-852, 857, 860-865, 869-871, 874-875, 880, 897, 903, 912, 916, 920, 1009, 1025-1027, 1041-1043, 1112, 1122, 1200, 1208, 1252-1253, 1280, 4909, 4934, 4946, 4948, 4953, 4960, 4970-4971, 5012, 5123, 8229, 9030, 9044, 9049, 9056, 9061, 9066, 13488, 17248, 17584, 25473, 25479, 25617, 25619, 28709

Tabela 674. IBM MQ for z/OS Suporte à conversão de CCSID (continuação)

CCSID	Converte para e de CCSIDS
424	37, 256, 420, 437, 500, 737, 775, 803, 819, 836, 850, 852, 856-857, 860-865, 916, 1112, 1122, 1200, 1208, 1252, 1255, 4932, 4946, 4948, 4952-4953, 4960, 5012, 5351, 8229, 8612, 9044, 9049, 9056, 13488, 16804, 17248, 17584, 28709
437	37, 256, 259, 273, 275, 277-278, 280, 284-285, 290, 297, 420, 423-424, 500, 737, 775, 813, 819, 833, 836, 838, 850, 852, 855, 857-858, 860-863, 865-866, 869-871, 874-875, 880, 897, 903, 905, 912, 914-916, 920-924, 1025-1027, 1040-1043, 1047, 1051, 1097, 1098, 1114-1115, 1126, 1140-1149, 1200, 1208, 1252, 1257, 1275, 1280-1281, 1283, 4386, 4909, 4929, 4932, 4934, 4946, 4948, 4951, 4953, 4970-4971, 5012, 5123, 5210-5211, 5348, 8229, 8482, 8612, 9025, 9030, 9044, 9049, 9061, 9066, 13121, 13488, 16804, 17584, 25473, 25479, 25617, 25619, 28709
500	37, 256, 273-275, 277-278, 280, 282, 284-285, 290, 297, 367, 420, 423-424, 437, 737, 775, 813, 819, 833, 836, 838, 850-852, 855-858, 860-866, 869-871, 874-875, 880, 891, 895, 897, 903-905, 912, 914-916, 920-924, 1004, 1009-1021, 1023, 1025-1027, 1040-1043, 1046-1047, 1051, 1088-1089, 1097, 1100-1107, 1112, 1114-1115, 1122, 1124-1126, 1129-1133, 1137, 1140-1149, 1200, 1208, 1250-1258, 1275, 1280-1283, 4386, 4909, 4929, 4932, 4934, 4946, 4948, 4951-4953, 4960, 4970-4971, 5012, 5123, 5142, 5210-5211, 5346, 5348, 8229, 8482, 8612, 9025, 9030, 9044, 9049, 9056, 9061, 9066, 9238, 13121, 13488, 16804, 17248, 17584, 25473, 25479, 25480, 25617, 25619, 25664, 28709
720	37, 420, 864, 1200, 1208, 1256, 4960, 8229, 8612, 9056, 13488, 16804, 17248, 17584, 28709
737	37, 256, 273, 277-278, 280, 284-285, 290, 297, 420, 423-424, 437, 500, 813, 833, 836, 838, 850, 869-871, 875, 880, 905, 1025-1027, 1097, 1200, 1208, 1252-1253, 1280, 4386, 4909, 4929, 4932, 4934, 4946, 4971, 5123, 8229, 8482, 8612, 9025, 9030, 9061, 13121, 13488, 16804, 17584, 28709
775	37, 256, 273, 277-278, 280, 284-285, 290, 297, 420, 423-424, 437, 500, 833, 836, 838, 850, 870-871, 875, 880, 905, 1025-1027, 1097, 1112, 1122, 1200, 1208, 1252, 1257, 4386, 4929, 4932, 4934, 4946, 4971, 5123, 8229, 8482, 8612, 9025, 9030, 13121, 13488, 16804, 17584, 28709
803	424, 819, 850, 856, 862, 916, 1200, 1208, 1252, 1255, 4946, 4952, 5012, 13488, 17584
806	1200, 1208, 13488, 17584
808	259, 858-859, 872, 923-924, 1140, 1148, 1153-1154, 1200, 1208, 5347, 5348, 13488, 17584
813	37, 273, 277-278, 280, 284-285, 297, 423, 437, 500, 737, 819, 838, 850, 852, 857, 860-861, 863, 869-871, 874-875, 880, 897, 903, 912, 916, 920, 1025-1027, 1041-1043, 1200, 1208, 1252-1253, 1280, 4909, 4934, 4946, 4948, 4953, 4970-4971, 5012, 5123, 5349, 8229, 9030, 9044, 9049, 9061, 9066, 13488, 17584, 25473, 25479, 25617, 25619, 28709

Tabela 674. IBM MQ for z/OS Suporte à conversão de CCSID (continuação)

CCSID	Converte para e de CCSIDS
819	37, 256, 273, 275, 277-278, 280, 284-285, 290, 297, 367, 420, 423-424, 437, 500, 803, 813, 833, 836, 838, 850, 852, 855, 857-858, 860-861, 863-866, 869-871, 874-875, 880, 897, 903, 905, 912, 914-916, 920-924, 1004, 1025-1027, 1041-1043, 1047, 1051, 1088-1089, 1097, 1098, 1112, 1114, 1122-1123, 1126, 1130, 1132, 1137, 1140-1149, 1200, 1208, 1250-1255, 1257-1258, 1275, 1280-1281, 1283, 4386, 4909, 4929, 4932, 4934, 4946, 4948, 4951, 4953, 4960, 4970-4971, 5012, 5123, 5210, 5346, 5348, 8229, 8482, 8612, 9025, 9030, 9044, 9049, 9056, 9061, 9066, 13121, 13488, 16804, 17248, 17584, 25473, 25479, 25617, 25619, 25664, 28709
833	37, 256, 273, 277-278, 280, 284-285, 290, 297, 367, 437, 500, 737, 775, 819, 836, 850, 852, 855, 857, 860-865, 870-871, 891, 1009, 1025-1027, 1040-1043, 1088, 1112, 1122, 1126, 1200, 1208, 1252, 4386, 4929, 4932, 4946, 4948, 4951, 4953, 4960, 5123, 8229, 8482, 9025, 9044, 9049, 9056, 13121, 13488, 17248, 17584, 25617, 25619, 25664, 28709
834	926, 951, 1200, 1208, 1362, 4930, 9026, 13488, 17584
835	927, 947, 1200, 1208, 4931, 9027, 13488, 17584, 21427
836	37, 256, 273, 277-278, 280, 284-285, 290, 297, 367, 424, 437, 500, 737, 775, 819, 833, 850, 852, 855, 857, 870-871, 875, 903, 1009, 1025-1027, 1040-1043, 1088, 1112, 1114-1115, 1122, 1200, 1208, 1252, 4386, 4929, 4932, 4946, 4948, 4951, 4953, 4971, 5123, 5210-5211, 8229, 8482, 9025, 9044, 9049, 13121, 13488, 17584, 25479, 25617, 25619, 25664, 28709
837	928, 1200, 1208, 1380, 1385, 4933, 13488, 17584
838	37, 256, 273, 277-278, 280, 284-285, 297, 423, 437, 500, 737, 775, 813, 819, 850, 852, 857, 860-865, 869-871, 874-875, 880, 897, 903, 912, 916, 920, 1025-1027, 1041-1043, 1112, 1122, 1200, 1208, 1252, 4909, 4934, 4946, 4948, 4953, 4960, 4970-4971, 5012, 5123, 8229, 9030, 9044, 9049, 9056, 9061, 9066, 13488, 17248, 17584, 25473, 25479, 25617, 25619, 28709
848	924, 1148, 1158, 1200, 1208, 5347, 13488, 17584
849	924, 1148, 1154, 1200, 1208, 5347, 13488, 17584
850	37, 256, 259, 273, 275, 277-278, 280, 284-285, 290, 297, 367, 420, 423-424, 437, 500, 737, 775, 803, 813, 819, 833, 836, 838, 852, 855-858, 860-866, 869-871, 874-875, 880, 897, 903, 905, 912, 914-916, 920-924, 1004, 1025-1027, 1040-1043, 1047, 1051, 1088-1089, 1097, 1098, 1100, 1112, 1114, 1122, 1126, 1130, 1132, 1140-1149, 1200, 1208, 1250-1257, 1275, 1280-1281, 1283, 4386, 4909, 4929, 4932, 4934, 4946, 4948, 4951-4953, 4960, 4970-4971, 5012, 5123, 5210, 5346, 5348, 8229, 8482, 8612, 9025, 9030, 9044, 9049, 9056, 9061, 9066, 13121, 13488, 16804, 17248, 17584, 25473, 25479, 25617, 25619, 25664, 28709
851	259, 423, 500, 875, 1200, 1208, 4971, 13488, 17584

Tabela 674. IBM MQ for z/OS Suporte à conversão de CCSID (continuação)

CCSID	Converte para e de CCSIDS
852	37, 256, 259, 273, 277-278, 280, 284-285, 290, 297, 420, 423-424, 437, 500, 813, 819, 833, 836, 838, 850, 855, 857, 860-861, 863, 869-871, 874-875, 880, 897, 903, 905, 912, 916, 920, 1025-1027, 1040-1043, 1047, 1088, 1097, 1200, 1208, 1250, 1252, 1282, 4386, 4909, 4929, 4932, 4934, 4946, 4948, 4951, 4953, 4970-4971, 5012, 5123, 5346, 8229, 8482, 8612, 9025, 9030, 9044, 9049, 9061, 9066, 13121, 13488, 16804, 17584, 25473, 25479, 25617, 25619, 25664, 28709
855	37, 259, 273, 277-278, 280, 284-285, 290, 297, 437, 500, 819, 833, 836, 850, 852, 857, 866, 870-871, 878, 880, 912, 915, 1025-1027, 1040-1043, 1088, 1200, 1208, 1250-1252, 1283, 4386, 4929, 4932, 4946, 4948, 4951, 4953, 5123, 5346, 5347, 8229, 8482, 9025, 9044, 9049, 13121, 13488, 17584, 25617, 25619, 25664, 28709
856	259, 273, 424, 500, 803, 850, 862, 916, 1200, 1208, 1255, 4946, 4952, 5012, 5351, 13488, 17584
857	37, 256, 259, 273, 277-278, 280, 284-285, 290, 297, 420, 423-424, 437, 500, 813, 819, 833, 836, 838, 850, 852, 855, 860-861, 863, 869-871, 874-875, 880, 897, 903, 905, 912, 916, 920, 1025-1027, 1040-1043, 1088, 1097, 1200, 1208, 1252, 1254, 1281, 4386, 4909, 4929, 4932, 4934, 4946, 4948, 4951, 4953, 4970-4971, 5012, 5123, 5350, 8229, 8482, 8612, 9025, 9030, 9044, 9049, 9061, 9066, 13121, 13488, 16804, 17584, 25473, 25479, 25617, 25619, 25664, 28709
858	37, 259, 273, 277-278, 280, 284-285, 297, 437, 500, 808, 819, 850, 860-861, 865, 871-872, 901-902, 923-924, 1047, 1051, 1140-1149, 1153-1157, 1160-1162, 1164, 1200, 1208, 1252, 1275, 4946, 5348, 8229, 13488, 17584, 28709
859	808, 872, 901-902, 1153-1157, 1160-1162, 1164, 1200, 1208, 13488, 17584
860	37, 256, 259, 273, 277-278, 280, 284-285, 290, 297, 420, 423-424, 437, 500, 813, 819, 833, 838, 850, 852, 857-858, 861, 863, 865, 869-871, 874-875, 880, 897, 903, 905, 912, 916, 920, 923-924, 1025-1027, 1041-1043, 1097, 1140, 1145-1146, 1148, 1200, 1208, 1252, 4386, 4909, 4929, 4934, 4946, 4948, 4953, 4970-4971, 5012, 5123, 5348, 8229, 8482, 8612, 9025, 9030, 9044, 9049, 9061, 9066, 13121, 13488, 16804, 17584, 25473, 25479, 25617, 25619, 28709
861	37, 256, 259, 273, 277-278, 280, 284-285, 290, 297, 420, 423-424, 437, 500, 813, 819, 833, 838, 850, 852, 857-858, 860, 863, 869-871, 874-875, 880, 897, 903, 905, 912, 916, 920, 923-924, 1025-1027, 1041-1043, 1097, 1148, 1149, 1200, 1208, 1252, 4386, 4909, 4929, 4934, 4946, 4948, 4953, 4970-4971, 5012, 5123, 5348, 8229, 8482, 8612, 9025, 9030, 9044, 9049, 9061, 9066, 13121, 13488, 16804, 17584, 25473, 25479, 25617, 25619, 28709
862	37, 256, 259, 273, 277-278, 280, 284-285, 290, 297, 420, 423-424, 437, 500, 803, 833, 838, 850, 856, 870-871, 875, 880, 905, 916, 1025-1027, 1097, 1200, 1208, 1252, 1255, 4386, 4929, 4934, 4946, 4952, 4971, 5012, 5123, 5351, 8229, 8482, 8612, 9025, 9030, 12712, 13121, 13488, 16804, 17584, 28709

Tabela 674. IBM MQ for z/OS Suporte à conversão de CCSID (continuação)

CCSID	Converte para e de CCSIDS
863	37, 256, 259, 273, 277-278, 280, 284-285, 290, 297, 420, 423-424, 437, 500, 813, 819, 833, 838, 850, 852, 857, 860-861, 865, 869-871, 874-875, 880, 897, 903, 905, 912, 916, 920, 1025-1027, 1041-1043, 1051, 1097, 1140-1149, 1200, 1208, 1252, 1275, 4386, 4909, 4929, 4934, 4946, 4948, 4953, 4970-4971, 5012, 5123, 5348, 8229, 8482, 8612, 9025, 9030, 9044, 9049, 9061, 9066, 13121, 13488, 16804, 17584, 25473, 25479, 25617, 25619, 28709
864	37, 256, 259, 273, 277-278, 280, 284-285, 290, 297, 420, 423-424, 500, 720, 819, 833, 838, 850, 870-871, 875, 880, 905, 918, 1008, 1025-1027, 1046, 1089, 1097, 1127, 1200, 1208, 1252, 1256, 4386, 4929, 4934, 4946, 4960, 4971, 5104, 5123, 5142, 5352, 8229, 8482, 8612, 9025, 9030, 9056, 9238, 13121, 13488, 16804, 17248, 17584, 28709
865	37, 256, 259, 273, 277-278, 280, 284-285, 290, 297, 420, 423-424, 437, 500, 819, 833, 838, 850, 858, 860, 863, 870-871, 875, 880, 905, 923-924, 1025-1027, 1097, 1142-1143, 1148, 1200, 1208, 1252, 4386, 4929, 4934, 4946, 4971, 5123, 5348, 8229, 8482, 8612, 9025, 9030, 13121, 13488, 16804, 17584, 28709
866	37, 256, 437, 500, 819, 850, 855, 870, 878, 880, 915, 1025, 1200, 1208, 1251-1252, 1283, 4946, 4951, 5347, 8229, 13488, 17584, 28709
867	259, 1153-1155, 1160, 1200, 1208, 4899, 5351, 9048, 12712, 13488, 17584
868	918, 1006, 1200, 1208, 13488, 17584
869	37, 256, 259, 273, 277-278, 280, 284-285, 297, 423, 437, 500, 737, 813, 819, 838, 850, 852, 857, 860-861, 863, 870-871, 874-875, 880, 897, 903, 912, 916, 920, 1025-1027, 1041-1043, 1200, 1208, 1252-1254, 1280, 4909, 4934, 4946, 4948, 4953, 4970-4971, 5012, 5123, 5349, 8229, 9030, 9044, 9049, 9061, 9066, 13488, 17584, 25473, 25479, 25617, 25619, 28709
870	37, 256, 273, 277-278, 280, 284-285, 290, 297, 423, 437, 500, 737, 775, 813, 819, 833, 836, 838, 850, 852, 855, 857, 860-866, 869, 871, 874-875, 880, 897, 903, 912, 915-916, 920, 1009, 1025-1027, 1040-1043, 1047, 1088, 1112, 1122, 1200, 1208, 1250, 1252, 1282, 4386, 4909, 4929, 4932, 4934, 4946, 4948, 4951, 4953, 4960, 4970-4971, 5012, 5123, 5346, 8229, 8482, 9025, 9030, 9044, 9049, 9056, 9061, 9066, 13121, 13488, 17248, 17584, 25473, 25479, 25617, 25619, 25664, 28709
871	37, 256, 273, 277-278, 280, 284-285, 290, 297, 367, 423, 437, 500, 737, 775, 813, 819, 833, 836, 838, 850, 852, 855, 857-858, 860-865, 869, 870, 874-875, 880, 897, 903, 912, 916, 920, 923-924, 1009, 1025-1027, 1040-1043, 1047, 1051, 1088, 1112, 1122, 1140-1149, 1200, 1208, 1252, 1275, 4386, 4909, 4929, 4932, 4934, 4946, 4948, 4951, 4953, 4960, 4970-4971, 5012, 5123, 5348, 8229, 8482, 9025, 9030, 9044, 9049, 9056, 9061, 9066, 13121, 13488, 17248, 17584, 25473, 25479, 25617, 25619, 25664, 28709
872	259, 808, 858-859, 923-924, 1140-1149, 1153-1155, 1200, 1208, 5347, 5348, 13488, 17584



Tabela 674. IBM MQ for z/OS Suporte à conversão de CCSID (continuação)

CCSID	Converte para e de CCSIDS
874	37, 259, 273, 277-278, 280, 284-285, 297, 423, 437, 500, 813, 819, 838, 850, 852, 857, 860-861, 863, 869-871, 875, 880, 897, 903, 912, 916, 920, 1025-1027, 1041-1043, 1200, 1208, 1252, 4909, 4934, 4946, 4948, 4953, 4970-4971, 5012, 5123, 8229, 9030, 9044, 9049, 9061, 9066, 13488, 17584, 25473, 25479, 25617, 25619, 28709
875	37, 256, 273, 277-278, 280, 284-285, 297, 367, 423, 437, 500, 737, 775, 813, 819, 836, 838, 850-852, 857, 860-865, 869-871, 874, 880, 897, 903, 912, 916, 920, 1009, 1025-1027, 1041-1043, 1047, 1088, 1112, 1122, 1200, 1208, 1252-1253, 1280, 4909, 4932, 4934, 4946, 4948, 4953, 4960, 4970-4971, 5012, 5123, 5349, 8229, 9030, 9044, 9049, 9056, 9061, 9066, 13488, 17248, 17584, 25473, 25479, 25617, 25619, 25664, 28709
878	855, 866, 880, 915, 1025, 1131, 1200, 1208, 1251, 1283, 4951, 5347, 13488, 17584
880	37, 256, 273, 277-278, 280, 284-285, 297, 423, 437, 500, 737, 775, 813, 819, 838, 850, 852, 855, 857, 860-866, 869-871, 874-875, 878, 897, 903, 912, 915-916, 920, 1009, 1025-1027, 1041-1043, 1112, 1122, 1200, 1208, 1251-1252, 1283, 4909, 4934, 4946, 4948, 4951, 4953, 4960, 4970-4971, 5012, 5123, 5347, 8229, 9030, 9044, 9049, 9056, 9061, 9066, 13488, 17248, 17584, 25473, 25479, 25617, 25619, 28709
891	500, 833, 1088, 1200, 1208, 4929, 9025, 13121, 13488, 17584, 25664
895	290, 500, 1027, 1041, 1200, 1208, 4386, 5123, 8482, 13488, 17584, 25617
896	290, 1027, 1041, 1200, 1208, 4386, 4992, 5123, 8482, 13488, 17584, 25617
897	37, 273, 277-278, 280, 284-285, 290, 297, 423, 437, 500, 813, 819, 838, 850, 852, 857, 860-861, 863, 869-871, 874-875, 880, 903, 912, 916, 920, 1025-1027, 1041-1043, 1200, 1208, 1252, 4386, 4909, 4934, 4946, 4948, 4953, 4970-4971, 5012, 5123, 8229, 8482, 9030, 9044, 9049, 9061, 9066, 13488, 17584, 25473, 25479, 25617, 25619, 28709
899	259
901	259, 858-859, 902, 923-924, 1140, 1148, 1156-1157, 1200, 1208, 5348, 5353, 13488, 17584
902	259, 858-859, 901, 923-924, 1140, 1148, 1156-1157, 1200, 1208, 5348, 5353, 13488, 17584
903.	37, 273, 277-278, 280, 284-285, 297, 423, 437, 500, 813, 819, 836, 838, 850, 852, 857, 860-861, 863, 869-871, 874-875, 880, 897, 912, 916, 920, 1025-1027, 1041-1043, 1115, 1200, 1208, 1252, 4909, 4932, 4934, 4946, 4948, 4953, 4970-4971, 5012, 5123, 5211, 8229, 9030, 9044, 9049, 9061, 9066, 13488, 17584, 25473, 25479, 25617, 25619, 28709
904	37, 500, 1114, 1200, 1208, 5210, 8229, 13488, 17584, 25480, 28709
905.	37, 256, 437, 500, 737, 775, 819, 850, 852, 857, 860-865, 920, 1026, 1112, 1122, 1200, 1208, 1252, 1254, 1281, 4946, 4948, 4953, 4960, 8229, 9044, 9049, 9056, 13488, 17248, 17584, 28709

Tabela 674. IBM MQ for z/OS Suporte à conversão de CCSID (continuação)

<b>CCSID</b>	<b>Converte para e de CCSIDS</b>
912	37, 273, 277-278, 280, 284-285, 297, 423, 437, 500, 813, 819, 838, 850, 852, 855, 857, 860-861, 863, 869-871, 874-875, 880, 897, 903, 916, 920, 1025-1027, 1041-1043, 1047, 1200, 1208, 1250, 1252, 1282, 4909, 4934, 4946, 4948, 4951, 4953, 4970-4971, 5012, 5123, 5346, 8229, 9030, 9044, 9049, 9061, 9066, 13488, 17584, 25473, 25479, 25617, 25619, 28709
914	37, 437, 500, 819, 850, 1200, 1208, 1252, 1257, 4946, 8229, 13488, 17584, 28709
915	37, 259, 437, 500, 819, 850, 855, 866, 870, 878, 880, 1025, 1131, 1200, 1208, 1251-1252, 1283, 4946, 4951, 5347, 8229, 13488, 17584, 28709
916	37, 273, 277-278, 280, 284-285, 297, 423-424, 437, 500, 803, 813, 819, 838, 850, 852, 856-857, 860-863, 869-871, 874-875, 880, 897, 903, 912, 920, 1025-1027, 1041-1043, 1200, 1208, 1252, 1255, 4909, 4934, 4946, 4948, 4952-4953, 4970-4971, 5012, 5123, 5351, 8229, 9030, 9044, 9049, 9061, 9066, 13488, 17584, 25473, 25479, 25617, 25619, 28709
918	864, 868, 1006, 1200, 1208, 4960, 9056, 13488, 17248, 17584
920	37, 273, 277-278, 280, 284-285, 297, 423, 437, 500, 813, 819, 838, 850, 852, 857, 860-861, 863, 869-871, 874-875, 880, 897, 903, 905, 912, 916, 1025-1026, 1200, 1208, 1252, 1254, 1281, 4909, 4934, 4946, 4948, 4953, 4970-4971, 5012, 5350, 8229, 9030, 9044, 9049, 9061, 9066, 13488, 17584, 25473, 25479, 28709
921	37, 437, 500, 819, 850, 922, 1112, 1122, 1200, 1208, 1252, 1257, 4946, 5353, 8229, 13488, 17584, 28709
922	37, 437, 500, 819, 850, 921, 1112, 1122, 1200, 1208, 1252, 1257, 4946, 5353, 8229, 13488, 17584, 28709
923	37, 273, 277-278, 280, 284-285, 297, 437, 500, 808, 819, 850, 858, 860-861, 865, 871-872, 901-902, 924, 1047, 1051, 1140-1149, 1153-1158, 1160-1162, 1164, 1200, 1208, 1252, 1275, 4946, 5348, 8229, 13488, 17584, 28709
924	37, 273, 277-278, 280, 284-285, 297, 437, 500, 808, 819, 848-850, 858, 860-861, 865, 871-872, 901-902, 923, 1047, 1051, 1140-1149, 1153-1157, 1160-1164, 1200, 1208, 1252, 1275, 4946, 5348, 8229, 13488, 17584, 28709
926	834, 951, 9026
927	835, 947, 1200, 1208, 4931, 9027, 13488, 17584, 21427
928	837, 1200, 1208, 1380, 13488, 17584
930	931-932, 939, 942-943, 1200, 1208, 1390, 1399, 5026, 5028, 5035, 5038-5039, 9122, 9124, 9131, 9135, 13218-13219, 13231, 13488, 17314, 17584, 25508, 25518, 29614, 33698-33700, 37796
931	930, 932, 939, 942-943, 1390, 1399, 5026, 5028, 5035, 5038-5039, 9122, 9124, 9131, 9135, 13218-13219, 13231, 17314, 25508, 25518, 29614, 33698-33700, 37796
932	930-931, 939, 942-943, 1200, 1208, 1390, 1399, 5026, 5028, 5035, 5038-5039, 9122, 9124, 9131, 9135, 13218-13219, 13231, 13488, 17314, 17584, 25508, 25518, 29614, 33698-33700, 37796

Tabela 674. IBM MQ for z/OS Suporte à conversão de CCSID (continuação)

<b>CCSID</b>	<b>Converte para e de CCSIDS</b>
933	934, 944, 949, 1200, 1208, 1363-1364, 5029, 5045, 5460, 9125, 9555, 13221, 13488, 13651, 17317, 17584, 25510, 25520, 25525, 29616, 29621, 33717, 37813
934	933, 949, 5029, 5045, 5460, 9125, 13221, 17317, 25510, 25525, 29621, 33717, 37813
935	936, 946, 1200, 1208, 1381, 1386, 1388, 5031, 5477, 5482, 5484, 9127, 13223, 13488, 17584, 25512
936	935, 946, 1381, 5031, 5477, 5484, 9127, 13223, 25512
937	938, 948, 950, 1200, 1208, 1370, 5033, 5046, 9142, 13488, 17584, 25514, 25524, 29620
938	937, 950, 1370, 5033, 5046, 9142, 25514
939	930-932, 942-943, 1200, 1208, 1390, 1399, 5026, 5028, 5035, 5038-5039, 9122, 9124, 9131, 9135, 13218-13219, 13231, 13488, 17314, 17584, 25508, 25518, 29614, 33698-33700, 37796
941	300-301, 1200, 1208, 1351, 4396, 8492, 13488, 16684, 17584
942	930-932, 939, 943, 1200, 1208, 1390, 1399, 5026, 5028, 5035, 5038-5039, 9122, 9124, 9131, 9135, 13218-13219, 13231, 13488, 17314, 17584, 25508, 25518, 29614, 33698-33700, 37796
943	930-932, 939, 942, 1200, 1208, 1390, 1399, 5026, 5028, 5035, 5038-5039, 9122, 9124, 9131, 9135, 13218-13219, 13231, 13488, 17314, 17584, 25508, 25518, 29614, 33698-33700, 37796
944	933, 949, 1200, 1208, 5029, 5045, 5460, 9125, 13221, 13488, 17317, 17584, 25520, 25525, 29616, 29621, 33717, 37813
946	935-936, 1200, 1208, 5031, 5484, 9127, 13223, 13488, 17584, 25512
947	835, 927, 1200, 1208, 4931, 9027, 13488, 17584, 21427
948	937, 950, 1200, 1208, 1370, 5033, 5046, 9142, 13488, 17584, 25524, 29620
949	933-934, 944, 1200, 1208, 1363-1364, 5029, 5045, 5460, 9125, 9555, 13221, 13488, 13651, 17317, 17584, 25510, 25520, 25525, 29616, 29621, 33717, 37813
950	937-938, 948, 1200, 1208, 1370, 5033, 5046, 9142, 13488, 17584, 25514, 25524, 29620
951	834, 926, 1200, 1208, 1362, 4930, 9026, 13488, 17584
1004	500, 819, 850, 1200, 1208, 4946, 13488, 17584
1006	868, 918, 1200, 1208, 13488, 17584
1008	420, 864, 1200, 1208, 4960, 5104, 8612, 9056, 13488, 16804, 17248, 17584
1009	37, 273, 277-278, 280, 284, 290, 297, 367, 423, 500, 833, 836, 870-871, 875, 880, 1025-1026, 1200, 1208, 4386, 4929, 4932, 4971, 8229, 8482, 9025, 13121, 13488, 17584, 28709
1010	500, 1200, 1208, 13488, 17584

Tabela 674. IBM MQ for z/OS Suporte à conversão de CCSID (continuação)

CCSID	Converte para e de CCSIDS
1011	500, 1200, 1208, 13488, 17584
1012	500, 1200, 1208, 13488, 17584
1013	500, 1140, 1200, 1208, 13488, 17584
1014	500, 1200, 1208, 13488, 17584
1015	500, 1200, 1208, 13488, 17584
1016	500, 1200, 1208, 13488, 17584
1017	500, 1200, 1208, 13488, 17584
1018	500, 1200, 1208, 13488, 17584
1019	500, 1200, 1208, 13488, 17584
1020	500
1021	500
1023	500
1025	37, 256, 273, 277-278, 280, 284-285, 290, 297, 423, 437, 500, 737, 775, 813, 819, 833, 836, 838, 850, 852, 855, 857, 860-866, 869-871, 874-875, 878, 880, 897, 903, 912, 915-916, 920, 1009, 1026-1027, 1040-1043, 1051, 1088, 1112, 1122, 1131, 1200, 1208, 1251-1252, 1283, 4386, 4909, 4929, 4932, 4934, 4946, 4948, 4951, 4953, 4960, 4970-4971, 5012, 5123, 5347, 8229, 8482, 9025, 9030, 9044, 9049, 9056, 9061, 9066, 13121, 13488, 17248, 17584, 25473, 25479, 25617, 25619, 25664, 28709
1026	37, 256, 273, 277-278, 280, 284-285, 290, 297, 367, 423, 437, 500, 737, 775, 813, 819, 833, 836, 838, 850, 852, 855, 857, 860-865, 869-871, 874-875, 880, 897, 903, 905, 912, 916, 920, 1009, 1025, 1027, 1040-1043, 1047, 1088, 1112, 1122, 1200, 1208, 1252, 1254, 1281, 4386, 4909, 4929, 4932, 4934, 4946, 4948, 4951, 4953, 4960, 4970-4971, 5012, 5123, 5350, 8229, 8482, 9025, 9030, 9044, 9049, 9056, 9061, 9066, 13121, 13488, 17248, 17584, 25473, 25479, 25617, 25619, 25664, 28709
1027	37, 256, 273, 277-278, 280, 284-285, 290, 297, 367, 423, 437, 500, 737, 775, 813, 819, 833, 836, 838, 850, 852, 855, 857, 860-865, 869-871, 874-875, 880, 895-897, 903, 912, 916, 1025-1026, 1040-1043, 1047, 1088, 1112, 1122, 1139, 1200, 1208, 1252, 4386, 4909, 4929, 4932, 4934, 4946, 4948, 4951, 4953, 4960, 4970-4971, 4992, 5012, 5123, 8229, 8482, 9025, 9030, 9044, 9049, 9056, 9061, 9066, 13121, 13488, 17248, 17584, 25473, 25479, 25617, 25619, 25664, 28709
1040	37, 273, 277-278, 280, 284-285, 290, 297, 437, 500, 833, 836, 850, 852, 855, 857, 870-871, 1025-1027, 1041-1043, 1088, 1200, 1208, 4386, 4929, 4932, 4946, 4948, 4951, 4953, 5123, 8229, 8482, 9025, 9044, 9049, 13121, 13488, 17584, 25617, 25619, 25664, 28709
1041	37, 273, 277-278, 280, 284-285, 290, 297, 367, 423, 437, 500, 813, 819, 833, 836, 838, 850, 852, 855, 857, 860-861, 863, 869-871, 874-875, 880, 895-897, 903, 912, 916, 1025-1027, 1040, 1042-1043, 1088, 1200, 1208, 1252, 4386, 4909, 4929, 4932, 4934, 4946, 4948, 4951, 4953, 4970-4971, 4992, 5012, 5123, 8229, 8482, 9025, 9030, 9044, 9049, 9061, 9066, 13121, 13488, 17584, 25473, 25479, 25617, 25619, 25664, 28709

Tabela 674. IBM MQ for z/OS Suporte à conversão de CCSID (continuação)

<b>CCSID</b>	<b>Converte para e de CCSIDS</b>
1042	37, 273, 277-278, 280, 284-285, 290, 297, 423, 437, 500, 813, 819, 833, 836, 838, 850, 852, 855, 857, 860-861, 863, 869-871, 874-875, 880, 897, 903, 912, 916, 1025-1027, 1040, 1041, 1043, 1088, 1200, 1208, 4386, 4909, 4929, 4932, 4934, 4946, 4948, 4951, 4953, 4970-4971, 5012, 5123, 8229, 8482, 9025, 9030, 9044, 9049, 9061, 9066, 13121, 13488, 17584, 25473, 25479, 25617, 25619, 25664, 28709
1043	37, 273, 277-278, 280, 284-285, 290, 297, 423, 437, 500, 813, 819, 833, 836, 838, 850, 852, 855, 857, 860-861, 863, 869-871, 874-875, 880, 897, 903, 912, 916, 1025-1027, 1040, 1041, 1042, 1088, 1114, 1200, 1208, 4386, 4909, 4929, 4932, 4934, 4946, 4948, 4951, 4953, 4970-4971, 5012, 5123, 5210, 8229, 8482, 9025, 9030, 9044, 9049, 9061, 9066, 13121, 13488, 17584, 25473, 25479, 25617, 25619, 25664, 28709
1046	420, 500, 864, 1089, 1127, 1200, 1208, 1256, 4960, 5142, 5352, 8612, 9056, 9238, 13488, 16804, 17248, 17584
1047	37, 273-275, 277-278, 280, 281, 282, 284-285, 290, 297, 437, 500, 819, 850, 852, 858, 870-871, 875, 912, 923-924, 1026-1027, 1140-1149, 1200, 1208, 1252, 1254, 4946, 4948, 5123, 8229, 8482, 13488, 17584, 28709
1051	37, 273, 277-278, 280, 284-285, 297, 437, 500, 819, 850, 858, 863, 871, 923-924, 1025, 1097, 1140-1149, 1200, 1208, 1252, 1275, 4946, 5348, 8229, 13488, 17584, 28709
1088	37, 273, 277-278, 280, 284-285, 290, 297, 367, 500, 819, 833, 836, 850, 852, 855, 857, 870-871, 875, 891, 1025-1027, 1040-1043, 1126, 1200, 1208, 4386, 4929, 4932, 4946, 4948, 4951, 4953, 4971, 5123, 8229, 8482, 9025, 9044, 9049, 13121, 13488, 17584, 25617, 25619, 25664, 28709
1089	420, 500, 819, 850, 864, 1046, 1127, 1200, 1208, 1256, 4946, 4960, 5142, 5352, 8612, 9056, 9238, 13488, 16804, 17248, 17584
1097	37, 437, 500, 737, 775, 819, 850, 852, 857, 860-865, 1051, 1098, 1112, 1122, 1200, 1208, 1252, 4946, 4948, 4953, 4960, 8229, 9044, 9049, 9056, 13488, 17248, 17584, 28709
1098	259, 420, 437, 819, 850, 1097, 1200, 1208, 1252, 4946, 8612, 13488, 16804, 17584
1100	37, 273, 277-278, 280, 284-285, 297, 500, 850, 4946, 8229, 28709
1101	500
1102	500
1103	500
1104	500
1105	500
1106	500
1107	500

Tabela 674. IBM MQ for z/OS Suporte à conversão de CCSID (continuação)

<b>CCSID</b>	<b>Converte para e de CCSIDS</b>
1112	37, 256, 273, 277-278, 280, 284-285, 290, 297, 420, 423-424, 500, 775, 819, 833, 836, 838, 850, 870-871, 875, 880, 905, 921-922, 1025-1027, 1097, 1122, 1200, 1208, 1252, 1257, 4386, 4929, 4932, 4934, 4946, 4971, 5123, 5353, 8229, 8482, 8612, 9025, 9030, 13121, 13488, 16804, 17584, 28709
1114	37, 437, 500, 819, 836, 850, 904, 1043, 1115, 1200, 1208, 4932, 4946, 5210-5211, 8229, 13488, 17584, 25480, 25619, 28709
1115	37, 367, 437, 500, 836, 903, 1114, 1200, 1208, 4932, 5210-5211, 8229, 13488, 17584, 25479, 28709
1122	37, 256, 273, 277-278, 280, 284-285, 290, 297, 420, 423-424, 500, 775, 819, 833, 836, 838, 850, 870-871, 875, 880, 905, 921-922, 1025-1027, 1097, 1112, 1200, 1208, 1252, 1257, 4386, 4929, 4932, 4934, 4946, 4971, 5123, 5353, 8229, 8482, 8612, 9025, 9030, 13121, 13488, 16804, 17584, 28709
1123	819, 1124-1125, 1148, 1200, 1208, 1251-1252, 1283, 5347, 13488, 17584
1124	37, 500, 1123, 1125, 1200, 1208, 1251, 1283, 5347, 8229, 13488, 17584, 28709
1125	500, 1123, 1124, 1200, 1208, 1251, 1283, 5347, 13488, 17584
1126	37, 367, 437, 500, 819, 833, 850, 1088, 1200, 1208, 1252, 4929, 4946, 8229, 9025, 13121, 13488, 17584, 25664, 28709
1127	420, 864, 1046, 1089, 1256, 4960, 5142, 8612, 9056, 9238, 16804, 17248
1129	500, 1130, 1200, 1208, 1258, 5354, 13488, 17584
1130	37, 500, 819, 850, 1129, 1200, 1208, 1252, 1258, 4946, 5354, 8229, 13488, 17584, 28709
1131	37, 500, 878, 915, 1025, 1200, 1208, 1251, 1283, 5347, 8229, 13488, 17584, 28709
1132	37, 500, 819, 850, 1133, 1200, 1208, 1252, 4946, 8229, 13488, 17584, 28709
1133	500, 1132, 1200, 1208, 13488, 17584
1137	37, 500, 819, 1200, 1208, 8229, 13488, 17584, 28709
1139	290, 1027, 4386, 5123, 8482
1140	37, 273, 277-278, 280, 284-285, 297, 437, 500, 808, 819, 850, 858, 860, 863, 871-872, 901-902, 923-924, 1013, 1047, 1051, 1141-1149, 1153-1157, 1160-1162, 1164, 1200, 1208, 1252, 1275, 4946, 5348, 8229, 13488, 17584, 28709
1141	37, 273, 277-278, 280, 284-285, 297, 437, 500, 819, 850, 858, 863, 871-872, 923-924, 1047, 1051, 1140, 1142-1149, 1153-1157, 1160-1162, 1200, 1208, 1252, 1275, 4946, 5348, 8229, 13488, 17584, 28709

Tabela 674. IBM MQ for z/OS Suporte à conversão de CCSID (continuação)

<b>CCSID</b>	<b>Converte para e de CCSIDS</b>
1142	37, 273, 277-278, 280, 284-285, 297, 437, 500, 819, 850, 858, 863, 865, 871-872, 923-924, 1047, 1051, 1140-1141, 1143-1149, 1153-1157, 1160-1162, 1200, 1208, 1252, 1275, 4946, 5348, 8229, 13488, 17584, 28709
1143	37, 273, 277-278, 280, 284-285, 297, 437, 500, 819, 850, 858, 863, 865, 871-872, 923-924, 1047, 1051, 1140-1142, 1144-1149, 1153-1157, 1160-1162, 1200, 1208, 1252, 1275, 4946, 5348, 8229, 13488, 17584, 28709
1144	37, 273, 277-278, 280, 284-285, 297, 437, 500, 819, 850, 858, 863, 871-872, 923-924, 1047, 1051, 1140-1143, 1145-1149, 1153-1157, 1160-1162, 1200, 1208, 1252, 1275, 4946, 5348, 8229, 13488, 17584, 28709
1145	37, 273, 277-278, 280, 284-285, 297, 437, 500, 819, 850, 858, 860, 863, 871-872, 923-924, 1047, 1051, 1140-1144, 1146-1149, 1153-1157, 1160-1162, 1200, 1208, 1252, 1275, 4946, 5348, 8229, 13488, 17584, 28709
1146	37, 273, 277-278, 280, 284-285, 297, 437, 500, 819, 850, 858, 860, 863, 871-872, 923-924, 1047, 1051, 1140-1145, 1147-1149, 1153-1157, 1160-1162, 1200, 1208, 1252, 1275, 4946, 5348, 8229, 13488, 17584, 28709
1147	37, 273, 277-278, 280, 284-285, 297, 437, 500, 819, 850, 858, 863, 871-872, 923-924, 1047, 1051, 1140-1146, 1148-1149, 1153-1157, 1160-1162, 1200, 1208, 1252, 1275, 4946, 5348, 8229, 13488, 17584, 28709
1148	37, 273, 277-278, 280, 284-285, 297, 437, 500, 808, 819, 848-850, 858, 860-861, 863, 865, 871-872, 901-902, 923-924, 1047, 1051, 1123, 1140-1147, 1149, 1153-1164, 1200, 1208, 1252, 1275, 4899, 4946, 5348, 5349, 8229, 12712, 13488, 17584, 28709
1149	37, 273, 277-278, 280, 284-285, 297, 437, 500, 819, 850, 858, 861, 863, 871-872, 923-924, 1047, 1051, 1140-1148, 1153-1157, 1160-1162, 1200, 1208, 1252, 1275, 4946, 5348, 8229, 13488, 17584, 28709
1153	808, 858-859, 867, 872, 923-924, 1140-1149, 1154-1157, 1160-1162, 1200, 1208, 5348, 9044, 13488, 17584
1154	808, 849, 858-859, 867, 872, 923-924, 1140-1149, 1153, 1155-1157, 1160-1162, 1200, 1208, 5347, 5348, 13488, 17584
1155	858-859, 867, 872, 923-924, 1140-1149, 1153-1154, 1156-1157, 1160-1162, 1200, 1208, 5348, 5350, 9049, 13488, 17584
1156	858-859, 901-902, 923-924, 1140-1149, 1153-1155, 1157, 1160, 1200, 1208, 5348, 5353, 12712, 13488, 17584
1157	858-859, 901-902, 923-924, 1140-1149, 1153-1156, 1160, 1200, 1208, 5348, 5353, 12712, 13488, 17584
1158	848, 923, 1148, 1200, 1208, 5347, 5348, 13488, 17584
1159	1148, 1200, 1208, 13488, 17584
1160	858-859, 867, 923-924, 1140-1149, 1153-1157, 1161-1162, 1200, 1208, 5348, 13488, 17584

Tabela 674. IBM MQ for z/OS Suporte à conversão de CCSID (continuação)

<b>CCSID</b>	<b>Converte para e de CCSIDS</b>
1161	259, 858-859, 923-924, 1140-1149, 1153-1155, 1160, 5348, 17584
1162	259, 858-859, 923-924, 1140-1149, 1153-1155, 1160, 5348, 17584
1163	924, 1148, 1164, 5354, 17584
1164	858-859, 923-924, 1140, 1148, 1163, 1200, 1208, 5348, 5354, 13488, 17584
1166	1200,1208,13488,17584
1200	37, 256, 259, 273, 275, 277-278, 280, 282, 284-285, 290, 293, 297, 300-301, 367, 420, 423-424, 437, 500, 720, 737, 775, 803, 806, 808, 813, 819, 833-838, 848-852, 855-872, 874-875, 878, 880, 891, 895-897, 901-905, 912, 914-916, 918, 920-924, 927-928, 930, 932-933, 935, 937, 939, 941-944, 946-951, 1004, 1006, 1008-1019, 1025-1027, 1040-1043, 1046-1047, 1051, 1088-1089, 1097-1098, 1112, 1114-1115, 1122-1126, 1129-1133, 1137, 1140-1149, 1153-1160, 1164, 1166, 1208, 1250-1258, 1275-1277, 1280-1285, 1351, 1362-1364, 1370-1371, 1374-1379, 1380-1381, 1385-1386, 1388, 1390, 1399, 4899, 4909, 4930, 4933, 4948, 4951-4952, 4960, 4971, 5012, 5039, 5104, 5123, 5142, 5210, 5346-5354, 8482, 8612, 9027, 9030, 9044, 9048-9049, 9056, 9061, 9066, 9238, 12712, 13121, 13218, 13488, 16684, 16804, 17248, 17584, 21427, 28709
1208	37, 256, 259, 273, 275, 277-278, 280, 282, 284-285, 290, 293, 297, 300-301, 367, 420, 423-424, 437, 500, 720, 737, 775, 803, 806, 808, 813, 819, 833-838, 848-852, 855-872, 874-875, 878, 880, 891, 895-897, 901-905, 912, 914-916, 918, 920-924, 927-928, 930, 932-933, 935, 937, 939, 941-944, 946-951, 1004, 1006, 1008-1019, 1025-1027, 1040-1043, 1046-1047, 1051, 1088-1089, 1097-1098, 1112, 1114-1115, 1122-1126, 1129-1133, 1137, 1140-1149, 1153-1160, 1164, 1166, 1200, 1250-1258, 1275-1277, 1280-1285, 1351, 1362-1364, 1370-1371, 1374-1379, 1380-1381, 1385-1386, 1388, 1390, 1399, 4899, 4909, 4930, 4933, 4948, 4951-4952, 4960, 4971, 5012, 5026, 5035, 5039, 5104, 5123, 5142, 5210, 5346-5354, 8482, 8612, 9027, 9030, 9044, 9048-9049, 9056, 9061, 9066, 9238, 12712, 13121, 13218, 13488, 16684, 16804, 17248, 17584, 21427, 28709
1250	37, 259, 273, 500, 819, 850, 852, 855, 870, 912, 1200, 1208, 1252, 1282, 4946, 4948, 4951, 5346, 8229, 9044, 13488, 17584, 28709
1251	37, 256, 259, 500, 819, 850, 855, 866, 878, 880, 915, 1025, 1123-1125, 1131, 1200, 1208, 1252, 1283, 4946, 4951, 5347, 8229, 13488, 17584, 28709
1252	37, 256, 259, 273, 275, 277-278, 280, 284-285, 290, 297, 420, 423-424, 437, 500, 737, 775, 803, 813, 819, 833, 836, 838, 850, 852, 855, 857-858, 860-866, 869-871, 874-875, 880, 897, 903, 905, 912, 914-916, 920-924, 1025-1027, 1041, 1047, 1051, 1097-1098, 1112, 1122-1123, 1126, 1130, 1132, 1140-1149, 1200, 1208, 1250-1251, 1254-1255, 1257, 1275, 1280-1281, 1283, 4386, 4909, 4929, 4932, 4934, 4946, 4948, 4951, 4953, 4960, 4970-4971, 5012, 5123, 5346, 5348, 8229, 8482, 8612, 9025, 9030, 9044, 9049, 9056, 9061, 9066, 13121, 13488, 16804, 17248, 17584, 25473, 25479, 25617, 28709
1253	37, 259, 423, 500, 737, 813, 819, 850, 869, 875, 1200, 1208, 1280, 4909, 4946, 4971, 5349, 8229, 9061, 13488, 17584, 28709



Tabela 674. IBM MQ for z/OS Suporte à conversão de CCSID (continuação)

<b>CCSID</b>	<b>Converte para e de CCSIDS</b>
1254	37, 259, 500, 819, 850, 857, 869, 905, 920, 1026, 1047, 1200, 1208, 1252, 1281, 4946, 4953, 5350, 8229, 9049, 9061, 13488, 17584, 28709
1255	37, 259, 424, 500, 803, 819, 850, 856, 862, 916, 1200, 1208, 1252, 1281, 4946, 4952, 5012, 5351, 8229, 13488, 17584, 28709
1256	259, 420, 500, 720, 850, 864, 1046, 1089, 1127, 1200, 1208, 4946, 4960, 5142, 5352, 8612, 9056, 9238, 13488, 16804, 17248, 17584
1257	37, 259, 437, 500, 775, 819, 850, 914, 921-922, 1112, 1122, 1200, 1208, 1252, 4946, 5353, 8229, 13488, 17584, 28709
1258	37, 259, 500, 819, 1129-1130, 1200, 1208, 5354, 8229, 13488, 17584, 28709
1275	37, 256, 273, 277-278, 280, 284-285, 297, 437, 500, 819, 850, 858, 863, 871, 923-924, 1051, 1140-1149, 1200, 1208, 1252, 4946, 5348, 8229, 13488, 17584, 28709
1276	1200, 1208, 13488, 17584
1277	1200, 1208, 13488, 17584
12 80	37, 423, 437, 500, 737, 813, 819, 850, 869, 875, 1200, 1208, 1252-1253, 4909, 4946, 4971, 5349, 8229, 9061, 13488, 17584, 28709
1281	37, 437, 500, 819, 850, 857, 905, 920, 1026, 1200, 1208, 1252, 1254-1255, 4946, 4953, 5350, 8229, 9049, 13488, 17584, 28709
1282	500, 852, 870, 912, 1200, 1208, 1250, 4948, 5346, 9044, 13488, 17584
1283	37, 437, 500, 819, 850, 855, 866, 878, 880, 915, 1025, 1123-1125, 1131, 1200, 1208, 1251-1252, 4946, 4951, 5347, 8229, 13488, 17584, 28709
1284	1200, 1208, 13488, 17584
1285	1200, 1208, 13488, 17584
1351	300-301, 941, 1200, 1208, 4396, 8492, 13488, 16684, 17584
1362	834, 951, 1200, 1208, 4930, 9026, 13488, 17584
1363	933, 949, 1200, 1208, 1364, 5029, 5045, 5460, 9125, 9555, 13221, 13488, 13651, 17317, 17584, 25525, 29621, 33717, 37813
1364	933, 949, 1200, 1208, 1363, 5029, 5045, 5460, 9125, 9555, 13221, 13488, 13651, 17317, 17584, 25525, 29621, 33717, 37813
1370	937-938, 948, 950, 1200, 1208, 1371, 5033, 5046, 9142, 13488, 17584, 25514, 25524, 29620
1371	1200, 1208, 1370, 13488, 17584
1374	1200, 1208
1375	1200, 1208
1376	1200, 1208
1377	1200, 1208
1378	1200, 1208
1379	1200, 1208

Tabela 674. IBM MQ for z/OS Suporte à conversão de CCSID (continuação)

<b>CCSID</b>	<b>Converte para e de CCSIDS</b>
1380	837, 928, 1200, 1208, 1385, 4933, 13488, 17584
1381	935-936, 1200, 1208, 1386, 1388, 5031, 5477, 5482, 5484, 9127, 13223, 13488, 17584, 25512
1385	837, 1200, 1208, 1380, 4933, 13488, 17584
1386	935, 1200, 1208, 1381, 1388, 5031, 5477, 5482, 5484, 9127, 13223, 13488, 17584
1388	935, 1200, 1208, 1381, 1386, 5031, 5477, 5482, 5484, 5488, 9127, 13223, 13488, 17584
1390	930-932, 939, 942-943, 1200, 1208, 1399, 5026, 5028, 5035, 5038-5039, 5055, 9122, 9124, 9131, 9135, 13218-13219, 13231, 13488, 17314, 17584, 25508, 25518, 29614, 33698-33700, 37796
1399	930-932, 939, 942-943, 1200, 1208, 1390, 5026, 5028, 5035, 5038-5039, 5050, 9122, 9124, 9131, 9135, 13218-13219, 13231, 13488, 17314, 17584, 25508, 25518, 29614, 33698-33700, 37796
4386	37, 256, 273, 277-278, 280, 284-285, 290, 297, 367, 437, 500, 737, 775, 819, 833, 836, 850, 852, 855, 857, 860-865, 870-871, 895-897, 1009, 1025-1027, 1040-1043, 1088, 1112, 1122, 1139, 1252, 4929, 4932, 4946, 4948, 4951, 4953, 4960, 4992, 5123, 8229, 8482, 9025, 9044, 9049, 9056, 13121, 17248, 25473, 25617, 25619, 25664, 28709
4396	300-301, 941, 1351, 8492, 16684
4899	867, 1148, 1200, 1208, 5351, 9048, 12712, 13488, 17584
4909	37, 273, 277-278, 280, 284-285, 297, 423, 437, 500, 737, 813, 819, 838, 850, 852, 857, 860-861, 863, 869-871, 874-875, 880, 897, 903, 912, 916, 920, 1025-1027, 1041-1043, 1200, 1208, 1252-1253, 1280, 4934, 4946, 4948, 4953, 4970-4971, 5012, 5123, 5349, 8229, 9030, 9044, 9049, 9061, 9066, 13488, 17584, 25473, 25479, 25617, 25619, 28709
4929	37, 256, 273, 277-278, 280, 284-285, 290, 297, 367, 437, 500, 737, 775, 819, 833, 836, 850, 852, 855, 857, 860-865, 870-871, 891, 1009, 1025-1027, 1040-1043, 1088, 1112, 1122, 1126, 1252, 4386, 4932, 4946, 4948, 4951, 4953, 4960, 5123, 8229, 8482, 9025, 9044, 9049, 9056, 13121, 17248, 25617, 25619, 25664, 28709
4930	834, 951, 1200, 1208, 1362, 9026, 13488, 17584
4931	835, 927, 947, 9027, 21427
4932	37, 256, 273, 277-278, 280, 284-285, 290, 297, 367, 424, 437, 500, 737, 775, 819, 833, 836, 850, 852, 855, 857, 870-871, 875, 903, 1009, 1025-1027, 1040-1043, 1088, 1112, 1114-1115, 1122, 1252, 4386, 4929, 4946, 4948, 4951, 4953, 4971, 5123, 5210-5211, 8229, 8482, 9025, 9044, 9049, 13121, 25479, 25617, 25619, 25664, 28709
4933	837, 1200, 1208, 1380, 1385, 13488, 17584
4934	37, 256, 273, 277-278, 280, 284-285, 297, 423, 437, 500, 737, 775, 813, 819, 838, 850, 852, 857, 860-865, 869-871, 874-875, 880, 897, 903, 912, 916, 920, 1025-1027, 1041-1043, 1112, 1122, 1252, 4909, 4946, 4948, 4953, 4960, 4970-4971, 5012, 5123, 8229, 9030, 9044, 9049, 9056, 9061, 9066, 17248, 25473, 25479, 25617, 25619, 28709

Tabela 674. IBM MQ for z/OS Suporte à conversão de CCSID (continuação)

CCSID	Converte para e de CCSIDS
4946	37, 256, 259, 273, 275, 277-278, 280, 284-285, 290, 297, 367, 420, 423-424, 437, 500, 737, 775, 803, 813, 819, 833, 836, 838, 850, 852, 855-858, 860-866, 869-871, 874-875, 880, 897, 903, 905, 912, 914-916, 920-924, 1004, 1025-1027, 1040-1043, 1047, 1051, 1088-1089, 1097-1098, 1100, 1112, 1114, 1122, 1126, 1130, 1132, 1140-1149, 1250-1257, 1275, 1280-1281, 1283, 4386, 4909, 4929, 4932, 4934, 4948, 4951-4953, 4960, 4970-4971, 5012, 5123, 5210, 5346, 5348, 8229, 8482, 8612, 9025, 9030, 9044, 9049, 9056, 9061, 9066, 13121, 16804, 17248, 25473, 25479, 25617, 25619, 25664, 28709
4948	37, 256, 259, 273, 277-278, 280, 284-285, 290, 297, 420, 423-424, 437, 500, 813, 819, 833, 836, 838, 850, 852, 855, 857, 860-861, 863, 869-871, 874-875, 880, 897, 903, 905, 912, 916, 920, 1025-1027, 1040-1043, 1047, 1088, 1097, 1200, 1208, 1250, 1252, 1282, 4386, 4909, 4929, 4932, 4934, 4946, 4951, 4953, 4970-4971, 5012, 5123, 5346, 8229, 8482, 8612, 9025, 9030, 9044, 9049, 9061, 9066, 13121, 13488, 16804, 17584, 25473, 25479, 25617, 25619, 25664, 28709
4951	37, 259, 273, 277-278, 280, 284-285, 290, 297, 437, 500, 819, 833, 836, 850, 852, 855, 857, 866, 870-871, 878, 880, 912, 915, 1025-1027, 1040-1043, 1088, 1200, 1208, 1250-1252, 1283, 4386, 4929, 4932, 4946, 4948, 4953, 5123, 5346, 5347, 8229, 8482, 9025, 9044, 9049, 13121, 13488, 17584, 25617, 25619, 25664, 28709
4952	259, 273, 424, 500, 803, 850, 856, 862, 916, 1200, 1208, 1255, 4946, 5012, 5351, 13488, 17584
4953	37, 256, 259, 273, 277-278, 280, 284-285, 290, 297, 420, 423-424, 437, 500, 813, 819, 833, 836, 838, 850, 852, 855, 857, 860-861, 863, 869-871, 874-875, 880, 897, 903, 905, 912, 916, 920, 1025-1027, 1040-1043, 1088, 1097, 1252, 1254, 1281, 4386, 4909, 4929, 4932, 4934, 4946, 4948, 4951, 4970-4971, 5012, 5123, 5350, 8229, 8482, 8612, 9025, 9030, 9044, 9049, 9061, 9066, 13121, 16804, 25473, 25479, 25617, 25619, 25664, 28709
4960	37, 256, 259, 273, 277-278, 280, 284-285, 290, 297, 420, 423-424, 500, 720, 819, 833, 838, 850, 864, 870-871, 875, 880, 905, 918, 1008, 1025-1027, 1046, 1089, 1097, 1127, 1200, 1208, 1252, 1256, 4386, 4929, 4934, 4946, 4971, 5104, 5123, 5142, 5352, 8229, 8482, 8612, 9025, 9030, 9056, 9238, 13121, 13488, 16804, 17248, 17584, 28709
4970	37, 259, 273, 277-278, 280, 284-285, 297, 423, 437, 500, 813, 819, 838, 850, 852, 857, 860-861, 863, 869-871, 874-875, 880, 897, 903, 912, 916, 920, 1025-1027, 1041-1043, 1252, 4909, 4934, 4946, 4948, 4953, 4971, 5012, 5123, 8229, 9030, 9044, 9049, 9061, 9066, 25473, 25479, 25617, 25619, 28709
4971	37, 256, 273, 277-278, 280, 284-285, 297, 367, 423, 437, 500, 737, 775, 813, 819, 836, 838, 850-852, 857, 860-865, 869-871, 874-875, 880, 897, 903, 912, 916, 920, 1009, 1025-1027, 1041-1043, 1047, 1088, 1112, 1122, 1200, 1208, 1252-1253, 1280, 4909, 4932, 4934, 4946, 4948, 4953, 4960, 4970, 5012, 5123, 5349, 8229, 9030, 9044, 9049, 9056, 9061, 9066, 13488, 17248, 17584, 25473, 25479, 25617, 25619, 25664, 28709
4992	290, 896, 1027, 1041, 4386, 5123, 8482, 25617

Tabela 674. IBM MQ for z/OS Suporte à conversão de CCSID (continuação)

<b>CCSID</b>	<b>Converte para e de CCSIDS</b>
5012	37, 273, 277-278, 280, 284-285, 297, 423-424, 437, 500, 803, 813, 819, 838, 850, 852, 856-857, 860-863, 869-871, 874-875, 880, 897, 903, 912, 916, 920, 1025-1027, 1041-1043, 1200, 1208, 1252, 1255, 4909, 4934, 4946, 4948, 4952-4953, 4970-4971, 5123, 5351, 8229, 9030, 9044, 9049, 9061, 9066, 13488, 17584, 25473, 25479, 25617, 25619, 28709
5026	930-932, 939, 942-943, 1390, 1399, 5028, 5035, 5038-5039, 9122, 9124, 9131, 9135, 1208, 13218-13219, 13231, 17314, 25508, 25518, 29614, 33698-33700, 37796
5028	930-932, 939, 942-943, 1390, 1399, 5026, 5035, 5038-5039, 9122, 9124, 9131, 9135, 13218-13219, 13231, 17314, 25508, 25518, 29614, 33698-33700, 37796
5029	933-934, 944, 949, 1363-1364, 5045, 5460, 9125, 9555, 13221, 13651, 17317, 25510, 25520, 25525, 29616, 29621, 33717, 37813
5031	935-936, 946, 1381, 1386, 1388, 5477, 5482, 5484, 9127, 13223, 25512
5033	937-938, 948, 950, 1370, 5046, 9142, 25514, 25524, 29620
5035	930-932, 939, 942-943, 1390, 1399, 5026, 5028, 5038-5039, 9122, 9124, 9131, 9135, 1208, 13218-13219, 13231, 17314, 25508, 25518, 29614, 33698-33700, 37796
5038	930-932, 939, 942-943, 1390, 1399, 5026, 5028, 5035, 5039, 9122, 9124, 9131, 9135, 13218-13219, 13231, 17314, 25508, 25518, 29614, 33698-33700, 37796
5039	930-932, 939, 942-943, 1200, 1208, 1390, 1399, 5026, 5028, 5035, 5038, 9122, 9124, 9131, 9135, 13218-13219, 13231, 13488, 17314, 17584, 25508, 25518, 29614, 33698-33700, 37796
5045	933-934, 944, 949, 1363-1364, 5029, 5460, 9125, 9555, 13221, 13651, 17317, 25510, 25520, 25525, 29616, 29621, 33717, 37813
5046	937-938, 948, 950, 1370, 5033, 9142, 25514, 25524, 29620
5104	420, 864, 1008, 1200, 1208, 4960, 8612, 9056, 13488, 16804, 17248, 17584
5123	290, 367, 423, 437, 819, 1027, 1041, 1047, 1140-1149, 1156, 1157, 1160, 1200, 1208, 1252, 4948, 5348, 8482, 13488
5142	420, 500, 864, 1046, 1089, 1127, 1200, 1208, 1256, 4960, 5352, 8612, 9056, 9238, 13488, 16804, 17248, 17584
5210	37, 437, 500, 819, 836, 850, 904, 1043, 1114-1115, 1200, 1208, 4932, 4946, 5211, 8229, 13488, 17584, 25480, 25619, 28709
5211	37, 367, 437, 500, 836, 903, 1114-1115, 4932, 5210, 8229, 25479, 28709
5346	37, 259, 273, 500, 819, 850, 852, 855, 870, 912, 1200, 1208, 1250, 1252, 1282, 4946, 4948, 4951, 8229, 9044, 13488, 17584, 28709
5347	808, 848-849, 855, 866, 872, 878, 880, 915, 1025, 1123-1125, 1131, 1154, 1158, 1200, 1208, 1251, 1283, 4951, 13488, 17584

Tabela 674. IBM MQ for z/OS Suporte à conversão de CCSID (continuação)

<b>CCSID</b>	<b>Converte para e de CCSIDS</b>
5348	37, 259, 273, 275, 277-278, 280, 284-285, 297, 437, 500, 808, 819, 850, 858, 860-861, 863, 865, 871-872, 901-902, 923-924, 1051, 1140-1149, 1153-1158, 1160-1162, 1164, 1200, 1208, 1252, 1275, 4946, 8229, 13488, 17584, 28709
5349	813, 869, 875, 1148, 1200, 1208, 1253, 1280, 4909, 4971, 9061, 13488, 17584
5350	857, 920, 1026, 1155, 1200, 1208, 1254, 1281, 4953, 9049, 13488, 17584
5351	424, 856, 862, 867, 916, 1200, 1208, 1255, 4899, 4952, 5012, 9048, 12712, 13488, 17584
5352	420, 864, 1046, 1089, 1200, 1208, 1256, 4960, 5142, 8612, 9056, 9238, 13488, 16804, 17248, 17584
5353	901-902, 921-922, 1112, 1122, 1156-1157, 1200, 1208, 1257, 13488, 17584
5354	1129-1130, 1163, 1164, 1200, 1208, 1258, 13488, 17584
5460	933-934, 944, 949, 1363-1364, 5029, 5045, 9125, 9555, 13221, 13651, 17317, 25510, 25520, 25525, 29616, 29621, 33717, 37813
5477	935-936, 1381, 1386, 1388, 5031, 5482, 5484, 9127, 13223, 25512
5482	935, 1381, 1386, 1388, 5031, 5477, 5484, 9127, 13223
5484	935-936, 946, 1381, 1386, 1388, 5031, 5477, 5482, 9127, 13223, 25512
5488	1388
8229	37, 256, 273, 275, 277-278, 280, 284-285, 290, 297, 367, 420, 423-424, 437, 500, 720, 737, 775, 813, 819, 833, 836, 838, 850, 852, 855, 857-858, 860-866, 869-871, 874-875, 880, 897, 903-905, 912, 914-916, 920-924, 1009, 1025-1027, 1040-1043, 1047, 1051, 1088, 1097, 1100, 1112, 1114-1115, 1122, 1124, 1126, 1130-1132, 1137, 1140-1149, 1250-1255, 1257-1258, 1275, 1280-1281, 1283, 4386, 4909, 4929, 4932, 4934, 4946, 4948, 4951, 4953, 4960, 4970-4971, 5012, 5123, 5210-5211, 5346, 5348, 8482, 8612, 9025, 9030, 9044, 9049, 9056, 9061, 9066, 13121, 16804, 17248, 25473, 25479, 25480, 25617, 25619, 25664, 28709
8482	37, 256, 273, 277-278, 280, 284-285, 290, 297, 367, 437, 500, 737, 775, 819, 833, 836, 850, 852, 855, 857, 860-865, 870-871, 895-897, 1009, 1025-1027, 1040-1043, 1047, 1088, 1112, 1122, 1139, 1200, 1208, 1252, 4386, 4929, 4932, 4946, 4948, 4951, 4953, 4960, 4992, 5123, 8229, 9025, 9044, 9049, 9056, 13121, 13488, 17248, 17584, 25473, 25617, 25619, 25664, 28709
8492	300-301, 941, 1351, 4396, 16684
8612	37, 256, 420, 424, 437, 500, 720, 737, 775, 819, 850, 852, 857, 860-865, 1008, 1046, 1089, 1098, 1112, 1122, 1127, 1200, 1208, 1252, 1256, 4946, 4948, 4953, 4960, 5104, 5142, 5352, 8229, 9044, 9049, 9056, 9238, 13488, 16804, 17248, 17584, 28709

Tabela 674. IBM MQ for z/OS Suporte à conversão de CCSID (continuação)

<b>CCSID</b>	<b>Converte para e de CCSIDS</b>
9025	37, 256, 273, 277-278, 280, 284-285, 290, 297, 367, 437, 500, 737, 775, 819, 833, 836, 850, 852, 855, 857, 860-865, 870-871, 891, 1009, 1025-1027, 1040-1043, 1088, 1112, 1122, 1126, 1252, 4386, 4929, 4932, 4946, 4948, 4951, 4953, 4960, 5123, 8229, 8482, 9044, 9049, 9056, 13121, 17248, 25617, 25619, 25664, 28709
9026	834, 926, 951, 1362, 4930
9027	835, 927, 947, 1200, 1208, 4931, 13488, 17584, 21427
9030	37, 256, 273, 277-278, 280, 284-285, 297, 423, 437, 500, 737, 775, 813, 819, 838, 850, 852, 857, 860-865, 869-871, 874-875, 880, 897, 903, 912, 916, 920, 1025-1027, 1041-1043, 1112, 1122, 1200, 1208, 1252, 4909, 4934, 4946, 4948, 4953, 4960, 4970-4971, 5012, 5123, 8229, 9044, 9049, 9056, 9061, 9066, 13488, 17248, 17584, 25473, 25479, 25617, 25619, 28709
9044	37, 256, 259, 273, 277-278, 280, 284-285, 290, 297, 420, 423-424, 437, 500, 813, 819, 833, 836, 838, 850, 852, 855, 857, 860-861, 863, 869-871, 874-875, 880, 897, 903, 905, 912, 916, 920, 1025-1027, 1040-1043, 1047, 1088, 1097, 1153, 1200, 1208, 1250, 1252, 1282, 4386, 4909, 4929, 4932, 4934, 4946, 4948, 4951, 4953, 4970-4971, 5012, 5123, 5346, 8229, 8482, 8612, 9025, 9030, 9049, 9061, 9066, 13121, 13488, 16804, 17584, 25473, 25479, 25617, 25619, 25664, 28709
9048	867, 1200, 1208, 4899, 5351, 12712, 13488, 17584
9049	37, 256, 259, 273, 277-278, 280, 284-285, 290, 297, 420, 423-424, 437, 500, 813, 819, 833, 836, 838, 850, 852, 855, 857, 860-861, 863, 869-871, 874-875, 880, 897, 903, 905, 912, 916, 920, 1025-1027, 1040-1043, 1088, 1097, 1155, 1200, 1208, 1252, 1254, 1281, 4386, 4909, 4929, 4932, 4934, 4946, 4948, 4951, 4953, 4970-4971, 5012, 5123, 5350, 8229, 8482, 8612, 9025, 9030, 9044, 9061, 9066, 13121, 13488, 16804, 17584, 25473, 25479, 25617, 25619, 25664, 28709
9056	37, 256, 259, 273, 277-278, 280, 284-285, 290, 297, 420, 423-424, 500, 720, 819, 833, 838, 850, 864, 870-871, 875, 880, 905, 918, 1008, 1025-1027, 1046, 1089, 1097, 1127, 1200, 1208, 1252, 1256, 4386, 4929, 4934, 4946, 4960, 4971, 5104, 5123, 5142, 5352, 8229, 8482, 8612, 9025, 9030, 9238, 13121, 13488, 16804, 17248, 17584, 28709
9061	37, 256, 259, 273, 277-278, 280, 284-285, 297, 423, 437, 500, 737, 813, 819, 838, 850, 852, 857, 860-861, 863, 869-871, 874-875, 880, 897, 903, 912, 916, 920, 1025-1027, 1041-1043, 1200, 1208, 1252-1254, 1280, 4909, 4934, 4946, 4948, 4953, 4970-4971, 5012, 5123, 5349, 8229, 9030, 9044, 9049, 9066, 13488, 17584, 25473, 25479, 25617, 25619, 28709
9066	37, 259, 273, 277-278, 280, 284-285, 297, 423, 437, 500, 813, 819, 838, 850, 852, 857, 860-861, 863, 869-871, 874-875, 880, 897, 903, 912, 916, 920, 1025-1027, 1041-1043, 1200, 1208, 1252, 4909, 4934, 4946, 4948, 4953, 4970-4971, 5012, 5123, 8229, 9030, 9044, 9049, 9061, 13488, 17584, 25473, 25479, 25617, 25619, 28709
9122	930-932, 939, 942-943, 1390, 1399, 5026, 5028, 5035, 5038-5039, 9124, 9131, 9135, 13218-13219, 13231, 17314, 25508, 25518, 29614, 33698-33700, 37796

Tabela 674. IBM MQ for z/OS Suporte à conversão de CCSID (continuação)

<b>CCSID</b>	<b>Converte para e de CCSIDS</b>
9124	930-932, 939, 942-943, 1390, 1399, 5026, 5028, 5035, 5038-5039, 9122, 9131, 9135, 13218-13219, 13231, 17314, 25508, 25518, 29614, 33698-33700, 37796
9125	933-934, 944, 949, 1363-1364, 5029, 5045, 5460, 9555, 13221, 13651, 17317, 25510, 25520, 25525, 29616, 29621, 33717, 37813
9127	935-936, 946, 1381, 1386, 1388, 5031, 5477, 5482, 5484, 13223, 25512
9131	930-932, 939, 942-943, 1390, 1399, 5026, 5028, 5035, 5038-5039, 9122, 9124, 9135, 13218-13219, 13231, 17314, 25508, 25518, 29614, 33698-33700, 37796
9135	930-932, 939, 942-943, 1390, 1399, 5026, 5028, 5035, 5038-5039, 9122, 9124, 9131, 13218-13219, 13231, 17314, 25508, 25518, 29614, 33698-33700, 37796
9142	937-938, 948, 950, 1370, 5033, 5046, 25514, 25524, 29620
9238	420, 500, 864, 1046, 1089, 1127, 1200, 1208, 1256, 4960, 5142, 5352, 8612, 9056, 13488, 16804, 17248, 17584
9555	933, 949, 1363-1364, 5029, 5045, 5460, 9125, 13221, 13651, 17317, 25525, 29621, 33717, 37813
12712	862, 867, 1148, 1156-1157, 1200, 1208, 4899, 5351, 9048, 13488, 17584
13121	37, 256, 273, 277-278, 280, 284-285, 290, 297, 367, 437, 500, 737, 775, 819, 833, 836, 850, 852, 855, 857, 860-865, 870-871, 891, 1009, 1025-1027, 1040-1043, 1088, 1112, 1122, 1126, 1200, 1208, 1252, 4386, 4929, 4932, 4946, 4948, 4951, 4953, 4960, 5123, 8229, 8482, 9025, 9044, 9049, 9056, 13488, 17248, 17584, 25617, 25619, 25664, 28709
13218	930-932, 939, 942-943, 1200, 1208, 1390, 1399, 5026, 5028, 5035, 5038-5039, 9122, 9124, 9131, 9135, 13219, 13231, 13488, 17314, 17584, 25508, 25518, 29614, 33698-33700, 37796
13219	930-932, 939, 942-943, 1390, 1399, 5026, 5028, 5035, 5038-5039, 9122, 9124, 9131, 9135, 13218, 13231, 17314, 25508, 25518, 29614, 33698-33700, 37796
13221	933-934, 944, 949, 1363-1364, 5029, 5045, 5460, 9125, 9555, 13651, 17317, 25510, 25520, 25525, 29616, 29621, 33717, 37813
13223	935-936, 946, 1381, 1386, 1388, 5031, 5477, 5482, 5484, 9127, 25512
13231	930-932, 939, 942-943, 1390, 1399, 5026, 5028, 5035, 5038-5039, 9122, 9124, 9131, 9135, 13218-13219, 17314, 25508, 25518, 29614, 33698-33700, 37796

Tabela 674. IBM MQ for z/OS Suporte à conversão de CCSID (continuação)

CCSID	Converte para e de CCSIDS
13488	37, 256, 259, 273, 275, 277-278, 280, 282, 284-285, 290, 293, 297, 300-301, 367, 420, 423-424, 437, 500, 720, 737, 775, 803, 806, 808, 813, 819, 833-838, 848-852, 855-872, 874-875, 878, 880, 891, 895-897, 901-905, 912, 914-916, 918, 920-924, 927-928, 930, 932-933, 935, 937, 939, 941-944, 946-951, 1004, 1006, 1008-1019, 1025-1027, 1040-1043, 1046-1047, 1051, 1088-1089, 1097-1098, 1112, 1114-1115, 1122-1126, 1129-1133, 1137, 1140-1149, 1153-1160, 1164, 1166, 1200, 1208, 1250-1258, 1275-1277, 1280-1285, 1351, 1362-1364, 1370-1371, 1380-1381, 1385-1386, 1388, 1390, 1399, 4899, 4909, 4930, 4933, 4948, 4951-4952, 4960, 4971, 5012, 5039, 5104, 5123, 5142, 5210, 5346-5354, 8482, 8612, 9027, 9030, 9044, 9048-9049, 9056, 9061, 9066, 9238, 12712, 13121, 13218, 16684, 16804, 17248, 17584, 21427, 28709
13651	933, 949, 1363-1364, 5029, 5045, 5460, 9125, 9555, 13221, 17317, 25525, 29621, 33717, 37813
16684	300-301, 941, 1200, 1208, 1351, 4396, 8492, 13488, 17584
16804	37, 256, 420, 424, 437, 500, 720, 737, 775, 819, 850, 852, 857, 860-865, 1008, 1046, 1089, 1098, 1112, 1122, 1127, 1200, 1208, 1252, 1256, 4946, 4948, 4953, 4960, 5104, 5142, 5352, 8229, 8612, 9044, 9049, 9056, 9238, 13488, 17248, 17584, 28709
17248	37, 256, 259, 273, 277-278, 280, 284-285, 290, 297, 420, 423-424, 500, 720, 819, 833, 838, 850, 864, 870-871, 875, 880, 905, 918, 1008, 1025-1027, 1046, 1089, 1097, 1127, 1200, 1208, 1252, 1256, 4386, 4929, 4934, 4946, 4960, 4971, 5104, 5123, 5142, 5352, 8229, 8482, 8612, 9025, 9030, 9056, 9238, 13121, 13488, 16804, 17584, 28709
17314	930-932, 939, 942-943, 1390, 1399, 5026, 5028, 5035, 5038-5039, 9122, 9124, 9131, 9135, 13218-13219, 13231, 25508, 25518, 29614, 33698-33700, 37796
17317	933-934, 944, 949, 1363-1364, 5029, 5045, 5460, 9125, 9555, 13221, 13651, 25510, 25520, 25525, 29616, 29621, 33717, 37813
17584	37, 256, 259, 273, 275, 277-278, 280, 282, 284-285, 290, 293, 297, 300-301, 367, 420, 423-424, 437, 500, 720, 737, 775, 803, 806, 808, 813, 819, 833-838, 848-852, 855-872, 874-875, 878, 880, 891, 895-897, 901-905, 912, 914-916, 918, 920-924, 927-928, 930, 932-933, 935, 937, 939, 941-944, 946-951, 1004, 1006, 1008-1019, 1025-1027, 1040-1043, 1046-1047, 1051, 1088-1089, 1097-1098, 1112, 1114-1115, 1122-1126, 1129-1133, 1137, 1140-1149, 1153-1160, 1164, 1166, 1200, 1208, 1250-1258, 1275-1277, 1280-1285, 1351, 1362-1364, 1370-1371, 1380-1381, 1385-1386, 1388, 1390, 1399, 4899, 4909, 4930, 4933, 4948, 4951-4952, 4960, 4971, 5012, 5039, 5104, 5123, 5142, 5210, 5346-5354, 8482, 8612, 9027, 9030, 9044, 9048-9049, 9056, 9061, 9066, 9238, 12712, 13121, 13218, 13488, 16684, 16804, 17248, 21427, 28709
21427	835, 927, 947, 1200, 1208, 4931, 9027, 13488, 17584
25473	37, 273, 277-278, 280, 284-285, 290, 297, 423, 437, 500, 813, 819, 838, 850, 852, 857, 860-861, 863, 869-871, 874-875, 880, 897, 903, 912, 916, 920, 1025-1027, 1041-1043, 1252, 4386, 4909, 4934, 4946, 4948, 4953, 4970-4971, 5012, 5123, 8229, 8482, 9030, 9044, 9049, 9061, 9066, 25479, 25617, 25619, 28709



Tabela 674. IBM MQ for z/OS Suporte à conversão de CCSID (continuação)

<b>CCSID</b>	<b>Converte para e de CCSIDS</b>
25479	37, 273, 277-278, 280, 284-285, 297, 423, 437, 500, 813, 819, 836, 838, 850, 852, 857, 860-861, 863, 869-871, 874-875, 880, 897, 903, 912, 916, 920, 1025-1027, 1041-1043, 1115, 1252, 4909, 4932, 4934, 4946, 4948, 4953, 4970-4971, 5012, 5123, 5211, 8229, 9030, 9044, 9049, 9061, 9066, 25473, 25617, 25619, 28709
25480	37, 500, 904, 1114, 5210, 8229, 28709
25508	930-932, 939, 942-943, 1390, 1399, 5026, 5028, 5035, 5038-5039, 9122, 9124, 9131, 9135, 13218-13219, 13231, 17314, 25518, 29614, 33698-33700, 37796
25510	933-934, 949, 5029, 5045, 5460, 9125, 13221, 17317, 25525, 29621, 33717, 37813
25512	935-936, 946, 1381, 5031, 5477, 5484, 9127, 13223
25514	937-938, 950, 1370, 5033, 5046, 9142
25518	930-932, 939, 942-943, 1390, 1399, 5026, 5028, 5035, 5038-5039, 9122, 9124, 9131, 9135, 13218-13219, 13231, 17314, 25508, 29614, 33698-33700, 37796
25520	933, 944, 949, 5029, 5045, 5460, 9125, 13221, 17317, 25525, 29616, 29621, 33717, 37813
25524	937, 948, 950, 1370, 5033, 5046, 9142, 29620
25525	933-934, 944, 949, 1363-1364, 5029, 5045, 5460, 9125, 9555, 13221, 13651, 17317, 25510, 25520, 29616, 29621, 33717, 37813
25617	37, 273, 277-278, 280, 284-285, 290, 297, 367, 423, 437, 500, 813, 819, 833, 836, 838, 850, 852, 855, 857, 860-861, 863, 869-871, 874-875, 880, 895-897, 903, 912, 916, 1025-1027, 1040-1043, 1088, 1252, 4386, 4909, 4929, 4932, 4934, 4946, 4948, 4951, 4953, 4970-4971, 4992, 5012, 5123, 8229, 8482, 9025, 9030, 9044, 9049, 9061, 9066, 13121, 25473, 25479, 25619, 25664, 28709
25619	37, 273, 277-278, 280, 284-285, 290, 297, 423, 437, 500, 813, 819, 833, 836, 838, 850, 852, 855, 857, 860-861, 863, 869-871, 874-875, 880, 897, 903, 912, 916, 1025-1027, 1040-1043, 1088, 1114, 4386, 4909, 4929, 4932, 4934, 4946, 4948, 4951, 4953, 4970-4971, 5012, 5123, 5210, 8229, 8482, 9025, 9030, 9044, 9049, 9061, 9066, 13121, 25473, 25479, 25617, 25664, 28709
25664	37, 273, 277-278, 280, 284-285, 290, 297, 367, 500, 819, 833, 836, 850, 852, 855, 857, 870-871, 875, 891, 1025-1027, 1040-1043, 1088, 1126, 4386, 4929, 4932, 4946, 4948, 4951, 4953, 4971, 5123, 8229, 8482, 9025, 9044, 9049, 13121, 25617, 25619, 28709

Tabela 674. IBM MQ for z/OS Suporte à conversão de CCSID (continuação)

CCSID	Converte para e de CCSIDS
28709	37, 256, 273, 275, 277-278, 280, 284-285, 290, 297, 367, 420, 423-424, 437, 500, 720, 737, 775, 813, 819, 833, 836, 838, 850, 852, 855, 857-858, 860-866, 869-871, 874-875, 880, 897, 903-905, 912, 914-916, 920-924, 1009, 1025-1027, 1040-1043, 1047, 1051, 1088, 1097, 1100, 1112, 1114-1115, 1122, 1124, 1126, 1130-1132, 1137, 1140-1149, 1200, 1208, 1250-1255, 1257-1258, 1275, 1280-1281, 1283, 4386, 4909, 4929, 4932, 4934, 4946, 4948, 4951, 4953, 4960, 4970-4971, 5012, 5123, 5210-5211, 5346, 5348, 8229, 8482, 8612, 9025, 9030, 9044, 9049, 9056, 9061, 9066, 13121, 13488, 16804, 17248, 17584, 25473, 25479, 25480, 25617, 25619, 25664
29614	930-932, 939, 942-943, 1390, 1399, 5026, 5028, 5035, 5038-5039, 9122, 9124, 9131, 9135, 13218-13219, 13231, 17314, 25508, 25518, 33698-33700, 37796
29616	933, 944, 949, 5029, 5045, 5460, 9125, 13221, 17317, 25520, 25525, 29621, 33717, 37813
29620	937, 948, 950, 1370, 5033, 5046, 9142, 25524
29621	933-934, 944, 949, 1363-1364, 5029, 5045, 5460, 9125, 9555, 13221, 13651, 17317, 25510, 25520, 25525, 29616, 33717, 37813
33698	930-932, 939, 942-943, 1390, 1399, 5026, 5028, 5035, 5038-5039, 9122, 9124, 9131, 9135, 13218-13219, 13231, 17314, 25508, 25518, 29614, 33699-33700, 37796
33699	930-932, 939, 942-943, 1390, 1399, 5026, 5028, 5035, 5038-5039, 9122, 9124, 9131, 9135, 13218-13219, 13231, 17314, 25508, 25518, 29614, 33698, 33700, 37796
33700	930-932, 939, 942-943, 1390, 1399, 5026, 5028, 5035, 5038-5039, 9122, 9124, 9131, 9135, 13218-13219, 13231, 17314, 25508, 25518, 29614, 33698-33699, 37796
33717	933-934, 944, 949, 1363-1364, 5029, 5045, 5460, 9125, 9555, 13221, 13651, 17317, 25510, 25520, 25525, 29616, 29621, 37813
37796	930-932, 939, 942-943, 1390, 1399, 5026, 5028, 5035, 5038-5039, 9122, 9124, 9131, 9135, 13218-13219, 13231, 17314, 25508, 25518, 29614, 33698-33700
37813	933-934, 944, 949, 1363-1364, 5029, 5045, 5460, 9125, 9555, 13221, 13651, 17317, 25510, 25520, 25525, 29616, 29621, 33717

## Suporte de conversão IBM i

Uma lista completa de CCSIDs e conversões suportadas pelo IBM i, pode ser localizada na publicação IBM i apropriada

As páginas de códigos suportadas são listadas em [Mapeamentos de CCSID suportados](#).

## Suporte de conversão Unicode

Algumas plataformas suportam a conversão de dados do usuário para ou da codificação Unicode. As duas formas de codificação Unicode suportadas são UTF-16 (CCSIDs 1200, 13488 e 17584) e UTF-8 (CCSID 1208). Você deve usar CCSIDs 1200 ou 1208, pois representam a versão Unicode mais recente suportada.

Os pares substitutos UTF-16 (um par de caracteres UTF-16 de 2 bytes no intervalo X'D800'até X'DFFF' que representam um ponto de código Unicode acima de U + FFFF) são suportados. Se um CCSID de destino não contiver um mapeamento para um ponto de código representado por um par substituto UTF-16 , o par de caracteres será convertido em um único caractere de substituição.

As sequências de caracteres de combinação são suportadas pelo IBM MQ Isso significa que, em alguns casos, um caractere pré-composto no CCSID de origem será convertido em uma sequência de caracteres de combinação no CCSID de destino ou em outro caminho.

**Nota:** O IBM MQ não suporta CCSIDs do gerenciador de filas UTF-16 , portanto, os dados do cabeçalho da mensagem não podem ser codificados em UTF-16

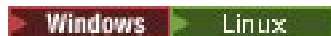
## Suporte IBM MQ AIX para Unicode



Na conversão IBM MQ for AIX para e de, os CCSIDs Unicode suportados (de preferência 1200 ou 1208) são suportados para os CCSIDs não Unicode na lista a seguir:

037  
273, 278, 280, 284, 285, 297  
423, 437  
500  
813, 819, 850, 852, 856, 857, 858, 860, 861, 865, 867, 869, 875, 878, 880  
901, 902, 912, 915, 916, 920, 923, 924, 932, 933, 935, 937, 938, 939, 942, 943, 948, 949, 950, 954, 964, 970  
1026, 1046, 1089  
1129, 1130, 1131, 1132, 1133, 1140, 1141, 1142, 1143, 1144, 1145, 1146 e 1147, 1149, 1153, 1156, 1157  
1200, 1208, 1250, 1251, 1253, 1254, 1258, 1280, 1281, 1282, 1283, 1284, 1285  
1363, 1364, 1381, 1383, 1386, 1388  
4899  
5026, 5035, 5050, 5346, 5347, 5348, 5349, 5350, 5351, 5352, 5353, 5354, 5488  
9044, 9048, 9449  
12712  
13488  
17584  
33722

## IBM MQ for Windows e Linux suporte para Unicode



Em IBM MQ for Windows e IBM MQ para Linux conversão para e de, os CCSIDs Unicode suportados (preferencialmente 1200 ou 1208) são suportados para os CCSIDs não Unicode na lista a seguir:

037,  
277, 278, 280, 284, 285, 290, 297  
300, 301  
420, 424, 437  
500  
813, 819, 833, 835, 836, 837 e 838, 850, 852, 855, 856, 857, 858, 860, 861, 862, 863, 864, 865, 866, 867, 867, 868, 869, 870, 871, 874 e 875, 878, 880, 891, 897  
901, 902, 903, 904, 912, 913<sup>“5”</sup> na página 1024, 915, 916, 918, 920, 921, 922, 923, 924, 927, 928, 930, 931<sup>“1”</sup> na página 1024, 932<sup>“2”</sup> na página 1024, 933, 935, 937, 938<sup>“3”</sup> na página 1024, 939, 941, 942, 943, 947, 948, 949, 950, 951, 954<sup>“4”</sup> na página 1024, 964, 970  
1006, 1025, 1026, 1027, 1040, 1041, 1042, 1043, 1046, 1047, 1051, 1088, 1089, 1097 e 1098

1112, 1114, 1115 e 1122, 1123, 1124, 1129, 1130, 1132, 1133, 1140, 1141, 1142, 1143, 1144, 1145, 1146, 1147, 1147, 1148, 1149, 1149, 1153, 1156, 1157  
1200, 1208, 1250, 1251 e 1252, 1253, 1254, 1255, 1256, 1257, 1258, 1275, 1280, 1281, 1282, 1283  
1363, 1364, 1374, 1375, 1376, 1377, 1378, 1379, 1380, 1381 e 1383, 1386, 1388  
4899  
5050, 5346, 5347, 5348; 5349; 5350; 5351; 5352; 5353; 5354; 5488<sup>“5”</sup> na página 1024  
9044, 9048, 9449  
12712  
13488  
17584  
33722<sup>“4”</sup> na página 1024

#### Notas:

1. 931 usa 939 para conversão.
2. 932 usa 942 para conversão.
3. 938 usa 948 para conversão.
4. 954 e 33722 usam 5050 para conversão.
5. Em Windows e Linux apenas

## Suporte IBM i para Unicode



Para obter detalhes sobre o suporte UNICODE, consulte a publicação IBM i apropriada relacionada ao sistema operacional.

## Suporte IBM MQ for z/OS para Unicode



Na conversão IBM MQ for z/OS para e de, os CCSIDs Unicode suportados (de preferência 1200 ou 1208) são suportados para os CCSIDs não Unicode na lista a seguir:

37  
256, 259, 273, 275, 277 e 278, 280, 282, 284, 285, 290, 293, 297  
300, 301, 367  
420, 423, 424, 437  
500  
720, 737, 775  
803, 806, 808, 813, 813, 819, 833, 834, 834, 835, 836, 837 e 838, 848, 849, 850, 851, 852, 855, 856, 857, 858, 859, 860, 861, 862, 863, 864, 865 e 866, 867, 868, 870, 871, 871, 872, 874, 875, 878, 880, 891, 895, 896, 897  
901, 902, 903, 904, 905, 912, 914, 915, 915, 916, 918, 920, 921, 922 e 923, 924, 927, 928, 930, 932, 933, 935, 937, 939, 941, 942, 943, 944, 946, 947, 948, 949, 950, 951  
1004, 1006, 1008, 1009, 1010, 1011 e 1012, 1013, 1014, 1015, 1016, 1017, 1018, 1019, 1025, 1026, 1027, 1040, 1041, 1042, 1043, 1046, 1047, 1051, 1088, 1089, 1097, 1098  
1112, 1114, 1115, 1122, 1123, 1124, 1125, 1126, 1129, 1130, 1131, 1132, 1133, 1137 e 1140, 1141, 1142, 1143, 1143, 1144, 1145, 1146, 1147, 1148, 1149, 1148, 1149, 1154, 1155, 1156, 1157, 1158, 1159, 1160, 1161, 1162, 1164  
1200, 1208, 1250, 1251, 1252 e 1253, 1254, 1255, 1256, 1257, 1258, 1275, 1276, 1277, 1280, 1281, 1282, 1283, 1284, 1285  
1351, 1362, 1363, 1364, 1370, 1371, 1380, 1381, 1385 e 1386, 1388, 1390, 1399  
4899, 4909, 4930, 4933, 4948, 4951 e 4952, 4960, 4971  
5012 5039 5104 5123 5142 5210 5346 5347 5348 5349 5350 5351 5352 5353 5354 5488

8482 8612  
9027 9030 9044 9048 9049 9056 9061 9066 9238 9449  
1166  
12712  
13121, 13218, 13488, 1374, 1375, 1376, 1377, 1378, 1379  
16684, 16804  
17248, 17584  
21427  
28709

## Padrões de codificação em plataformas de 64 bits

Use estas informações para aprender sobre padrões de codificação em plataformas de 64 bits e os tipos de dados preferenciais..

### Tipos de dados preferenciais

Esses tipos nunca mudam de tamanho e estão disponíveis nas plataformas IBM MQ de 32 bits e de 64 bits:

Tabela 675. Nomes e comprimentos do tipo de dados

Nome	Tamanho
MQLONG	4 bytes
MQULONG	4 bytes
MQINT32	4 bytes
MQUINT32	4 bytes
MQINT64	8 bytes
MQUINT64	8 bytes

### Tipos de dados padrão no AIX, Linux, and Windows .

Aprenda sobre os tipos de dados padrão em aplicativos AIX and Linux de 32 e 64 bits AIX, Linux, and Windows .

### Aplicativos AIX and Linux de 32 bits




Tabela 676. Nomes e comprimentos de tipo de dados para aplicativos AIX and Linux de 32 bits

Nome	Tamanho
char	1 byte
short	2 bytes
int	4 bytes
long	4 bytes
float	4 bytes
double	8 bytes
long double	8 bytes
ponteiro	4 bytes

Tabela 676. Nomes e comprimentos de tipo de dados para aplicativos AIX and Linux de 32 bits (continuação)

Nome	Tamanho
ptrdiff_t	4 bytes
size_t	4 bytes
time_t	4 bytes
clock_t	4 bytes
wchar_t	4 bytes


 Observe que em AIX um wchar\_t é 2 bytes.

### Aplicativos AIX and Linux de 64 bits

Tabela 677. Nomes e comprimentos de tipo de dados para aplicativos AIX and Linux de 64 bits

Nome	Tamanho
char	1 byte
short	2 bytes
int	4 bytes
long	8 bytes
float	4 bytes
double	8 bytes
long double	8 bytes
ponteiro	8 bytes
ptrdiff_t	8 bytes
size_t	8 bytes
time_t	8 bytes
clock_t	4 bytes
wchar_t	4 bytes

 Observe que em AIX um wchar\_t é 2 bytes.

### Windows Aplicativos de 64 bits



Tabela 678. Nomes e comprimentos de tipos de dados para aplicativos Windows de 64 bits

Nome	Tamanho
char	1 byte
short	2 bytes
int	4 bytes
long	4 bytes
float	4 bytes

Tabela 678. Nomes e comprimentos de tipos de dados para aplicativos Windows de 64 bits (continuação)

Nome	Tamanho
double	8 bytes
long double	8 bytes
ponteiro	8 bytes
	Note que todos os ponteiros têm 8 bytes.
ptrdiff_t	8 bytes
size_t	8 bytes
time_t	8 bytes
clock_t	4 bytes
wchar_t	2 bytes
WORD	2 bytes
DWORD	4 bytes
Identificador	8 bytes
HFile	4 bytes

## Considerações de codificação no Windows

### Windows

#### HANDLE hf;

Uso

```
hf = CreateFile((LPCTSTR) FileName,  
               Access,  
               ShareMode,  
               xihSecAttsNTRestrict,  
               Create,  
               AttrAndFlags,  
               NULL);
```

Não utilizar

```
HFILE hf;  
hf = (HFILE) CreateFile((LPCTSTR) FileName,  
                       Access,  
                       ShareMode,  
                       xihSecAttsNTRestrict,  
                       Create,  
                       AttrAndFlags,  
                       NULL);
```

como isso produz um erro.

#### size\_t len fgets

Uso

```
size_t len  
while (fgets(string1, (int) len, fp) != NULL)  
len = strlen(buffer);
```

Não utilizar

```
int len;
while (fgets(string1, len, fp) != NULL)
len = strlen(buffer);
```

## printf

Uso

```
printf("My struc pointer: %p", pMyStruc);
```

Não utilizar

```
printf("My struc pointer: %x", pMyStruc);
```

Se você precisar de saída hexadecimal, terá que imprimir os 4 bytes superior e inferior separadamente

## char \* ptr

Uso

```
char * ptr1;
char * ptr2;
size_t bufLen;

bufLen = ptr2 - ptr1;
```

Não utilizar

```
char *ptr1;
char *ptr2;
UINT32 bufLen;

bufLen = ptr2 - ptr1;
```

## alignBytes

Uso

```
alignBytes = (unsigned short) ((size_t) address % 16);
```

Não utilizar

```
void *address;
unsigned short alignBytes;

alignBytes = (unsigned short) ((UINT32) address % 16);
```

## len

Uso

```
len = (UINT32) ((char *) address2 - (char *) address1);
```

Não utilizar

```
void *address1;
void *address2;
UINT32 len;

len = (UINT32) ((char *) address2 - (char *) address1);
```



## sscanf

### Uso

```
MQLONG SBCSprt;  
sscanf(line, "%d", &SBCSprt);
```

### Não utilizar

```
MQLONG SBCSprt;  
sscanf(line, "%1d", &SBCSprt);
```

%1d tenta colocar um tipo de 8 bytes em um tipo de 4 bytes; use %1 apenas se estiver lidando com um tipo de dados long real. MQLONG, UINT32 e INT32 são definidos para serem quatro bytes, o mesmo que um int em todas as plataformas IBM MQ :

## IBM i IBM i Referência de programação de aplicativos (ILE/RPG)

### Programação de aplicativos para o IBM i

Use estas informações para ajudar a desenvolver aplicativos para o IBM i.

- [“Descrições de tipo de dados em IBM i ..” na página 1030](#)
- [“Chamadas de função em IBM i .” na página 1295](#)
- [“Atributos de objetos no IBM i” na página 1416](#)
- [“Aplicativos” na página 1464](#)
- [“Códigos de retorno para IBM i \(ILE RPG\)” na página 1477](#)
- [“Regras para validar opções MQI para IBM i \(ILE RPG\)” na página 1479](#)
- [“Codificações da máquina em IBM i” na página 1481](#)
- [“Opções de relatório e sinalizações de mensagem no IBM i” na página 1484](#)

### Descontinuação do modo de compatibilidade de aplicativos RPG e COBOL no IBM i

#### IBM i

No IBM MQ for IBM i 9.0, o produto não fornece mais suporte para aplicativos RPG ou COBOL que usam a ligação dinâmica conhecida como modo de compatibilidade. Esse modo de operação era necessário para aplicativos gravados antes do MQSeries 5.1 e versões subsequentes do produto forneciam um ambiente de tempo de execução compatível para esses aplicativos, embora os copybooks necessários para compilá-los tivessem sido removidos no IBM WebSphere MQ 6.0. A ligação dinâmica (modo de compatibilidade) era fornecida pelos programas a seguir na biblioteca QMQM, que foram removidos no IBM MQ for IBM i 9.0:

- AMQVSTUB
- AMQZSTUB
- QMQM
- MQCLOSE
- MQCONN
- MQDISC
- MQGET
- MQINQ
- MQOPEN
- MQPUT

- MQPUT1
- MQSET

A partir da IBM MQ for IBM i 9.0, aplicativos que usam esse modo de compatibilidade de operação precisam ser recompilados para usar as chamadas MQ de limite estático fornecidas pelos programas de serviços LIBMQM e LIBMQM\_R. Os programas de amostra, como AMQ3PUT4 e AMQ3GET4, mostram como usar esse modelo de programação. Para obter mais informações sobre como usar essas chamadas MQ, consulte [IBM i Application Programming Reference \(ILE/RPG\)](#).

#### Notas:

- É necessário recodificar os aplicativos, que usam atualmente a interface CALL 'QMQM', para usar o programa de serviços LIBMQM.

Os objetos de programa e programas de serviços na lista anterior, por exemplo, QMQM, MQCONN, MQPUT, AMQVSTUB e AMQZSTUB, são removidos no IBM MQ for IBM i 9.0 e os aplicativos que foram codificados para usar o modo de compatibilidade deixam de funcionar.

- Se os aplicativos estiverem ligados ao programa de serviço LIBMQM em IBM MQ for IBM i 8.0, não será necessário recompilar ou vincular novamente esses aplicativos em IBM MQ for IBM i 9.0 ou posteriormente.
- Não é possível instalar mais de uma versão do IBM MQ for IBM i na mesma partição..

Para descobrir se o programa RPG ou COBOL usa o modo de compatibilidade, use o comando **DSPPGMREF** (Exibir referências do programa) para exibir os programas externos chamados pelo programa de aplicativo. Se houver referências aos programas listados nesta seção, o programa não será executado em IBM MQ for IBM i 9.0 ou posterior. O exemplo a seguir da saída de **DSPPGMREF** mostra três objetos de programa que foram descontinuados, MQCONN, MQOPEN e MQCLOSE:

```

Program . . . . . : MYAPPPGM
Library . . . . . : MYLIB
Text 'description'. . . . . : ILE/COBOL SAMPLE PUT TO QUEUE (MQPUT)
Number of objects referenced . . . . . : 5
Object . . . . . : MQCONN
Library . . . . . : *LIBL
Object type . . . . . : *PGM
Object . . . . . : MQOPEN
Library . . . . . : *LIBL
Object type . . . . . : *PGM
Object . . . . . : MQCLOSE
Library . . . . . : *LIBL
Object type . . . . . : *PGM

```

Esses programas devem ser recompilados usando o método Bound Procedural Call descrito em [Preparando programas COBOL no IBM i](#).

Se você tentar executar um programa de aplicativo no IBM MQ for IBM i 9.0 ou mais recente que usa o modo de compatibilidade, o primeiro erro mais comumente visto será um MCH3401 tentando chamar o programa MQCONN ou QMQM

#### Tarefas relacionadas

[Desenvolvendo Aplicativos](#)

## IBM i Descrições de tipo de dados em IBM i ..

Esta coleção de tópicos fornece descrições dos tipos de dados usados em programação do IBM i

### Convenções usadas na descrição de tipos de dados

Para cada tipo de dados elementares, essas informações fornecem uma descrição de seu uso, em uma forma independente da linguagem de programação. Isto é seguido por declarações típicas na versão ILE da linguagem de programação RPG. As definições dos tipos de dados elementares são incluídas aqui para fornecer consistência. O RPG usa ' D' especificações onde os campos de trabalho podem ser declarados usando os atributos que você precisa. É possível, no entanto, fazer isso nas especificações de cálculo em que o campo é usado

Para usar os tipos de dados elementares, você cria:

- Um membro /COPY que contém todos os tipos de dados ou
- Uma estrutura de dados externos (PF) contendo todos os tipos de dados. Em seguida, é necessário especificar seus campos de trabalho com os atributos 'LIKE' do campo de tipo de dados apropriado

Os benefícios da segunda opção são que as definições podem ser usadas como um 'FIELD REFERENCE FILE' para outros objetos IBM i . Se uma definição de tipo de dado IBM MQ for alterada, é relativamente simples recriar esses objetos.

## Tipos de dados elementares

Todos os outros tipos de dados descritos nesta seção equivalem diretamente a esses tipos de dados elementares ou a agregações desses tipos de dados elementares (matrizes ou estruturas).

<i>Tabela 679. Tipos de dados elementares</i>	
<b>Tipo de Dados</b>	<b>Representação</b>
MQBOOL	Número inteiro assinado de 10 dígitos
MQBYTE	Campo alfanumérico de 1 byte
MQBYTE16	campo alfanumérico de 16 bytes
MQBYTE24	Campo alfanumérico de 24 bytes
MQBYTE32	Campo alfanumérico de 32 bytes
MQBYTE64	Campo alfanumérico de 64 bytes
MQCHAR	Campo alfanumérico de 1 byte
MQCHAR4	Campo alfanumérico de 4 bytes
MQCHAR8	Campo alfanumérico de 8 bytes
MQCHAR12	campo alfanumérico de 12 bytes
MQCHAR16	campo alfanumérico de 16 bytes
MQCHAR20	Campo alfanumérico de 20 bytes
MQCHAR28	Campo alfanumérico de 28 bytes.
MQCHAR32	Campo alfanumérico de 32 bytes
MQCHAR48	campo alfanumérico de 48 bytes
MQCHAR64	Campo alfanumérico de 64 bytes
MQCHAR128	Campo alfanumérico de 128 bytes
MQCHAR256	Campo alfanumérico de 256 bytes
MQFLOAT32	Número de vírgula flutuante de 4 bytes
MQFLOAT64	Número de vírgula flutuante de 8 bytes
MQHCONFIG	Identificador de Configuração.
MQHCONN	Número inteiro assinado de 10 dígitos
MQHMSG	Identificador de mensagem que fornece acesso a uma mensagem
MQHOBJ	Número inteiro assinado de 10 dígitos
MQINT8	Número inteiro assinado de 8 bits

Tabela 679. Tipos de dados elementares (continuação)

<b>Tipo de Dados</b>	<b>Representação</b>
MQINT16	Número inteiro assinado de 16 bits
MQINT32	Número inteiro assinado de 32 bits
MQINT64	Número inteiro assinado de 64 bits
MQLONG	Número inteiro assinado de 32 bits
MQPID	Identificador de Processo
MQPTR	Ponteiro
MQTID	Identificador de encadeamento
MQUINT8	Número inteiro não assinado de 8 bits
MQUINT16	Número inteiro não assinado de 16 bits
MQUINT32	Número inteiro não assinado de 32 bits
MQUINT64	Número inteiro não assinado de 64 bits
MQULONG	Número inteiro não assinado de 32 bits
PMQACH	Ponteiro para uma estrutura de dados do tipo MQACH.
PMQAIR	Ponteiro para uma estrutura de dados do tipo MQAIR.
PMQAXC	Ponteiro para uma estrutura de dados do tipo MQAXC.
PMAXP	Ponteiro para uma estrutura de dados do tipo MAXP
PMQBMHO	Ponteiro para uma estrutura de dados do tipo MQBMHO
PMQBO	Ponteiro para uma estrutura de dados do tipo MQBO
PMQBOOL	Ponteiro para dados do tipo MQBOOL
PMQBYTE	Ponteiro para dados do tipo MQBYTE
PMQBYTE <sub>n</sub>	Ponteiro para dados do tipo MQBYTE <sub>n</sub>
PMQCBC	Ponteiro para uma estrutura de dados do tipo MQCBC
PMQCBD	Ponteiro para uma estrutura de dados do tipo MQCB
PMQCHAR	Ponteiro para uma estrutura de dados do tipo MQCHAR
PMQCHARV	Ponteiro para uma estrutura de dados do tipo MQCHARV..
PMQCHAR <sub>n</sub>	Ponteiro para dados do tipo MQCHAR <sub>n</sub>
PMQCIH	Ponteiro para uma estrutura de dados do tipo MQCIH

Tabela 679. Tipos de dados elementares (continuação)

<b>Tipo de Dados</b>	<b>Representação</b>
PMQCMHO	Ponteiro para uma estrutura de dados do tipo MQCMHO
PMQCNO	Ponteiro para uma estrutura de dados do tipo MQCN
PMQCSP	Ponteiro para uma estrutura de dados do tipo MQCSP
PMQCTLO	Ponteiro para uma estrutura de dados do tipo MQCTLO
PMQDH	Ponteiro para uma estrutura de dados do tipo MQDH
PMQDHO	Ponteiro para uma estrutura de dados do tipo MQDHO
PMQDLH	Ponteiro para uma estrutura de dados do tipo MQDLH
PMQDMHO	Ponteiro para uma estrutura de dados do tipo MQDMHO
PMQDMPO	Ponteiro para uma estrutura de dados do tipo MQDMPO
PMQEPH	Ponteiro para uma estrutura de dados do tipo MQEPH
PMQFLOAT32	Ponteiro para dados do tipo MQFLOAT32
PMQFLOAT64	Ponteiro para dados do tipo MQFLOAT64
PMQFUNC	Ponteiro para uma função
PMQGMO	Ponteiro para uma estrutura de dados do tipo MQGMO..
PMQHCONFIG	Ponteiro para dados do tipo MQHCONFIG
PMQHCONN	Ponteiro para dados do tipo MQHCONN
PMQHMSG	Ponteiro para dados do tipo MQHMSG
PMQHOBJ	Ponteiro para dados do tipo MQHOBJ
PMQIIH	Ponteiro para uma estrutura de dados do tipo MQIIH..
PMQIMPO	Ponteiro para uma estrutura de dados do tipo MQIMPO
PMQINT8	Ponteiro para dados do tipo MQINT8
PMQINT16	Ponteiro para dados do tipo MQINT16
PMQINT32	Ponteiro para dados do tipo MQINT32
PMQINT64	Ponteiro para dados do tipo MQINT64
PMQLONG	Ponteiro para dados do tipo MQLONG

Tabela 679. Tipos de dados elementares (continuação)

<b>Tipo de Dados</b>	<b>Representação</b>
PMQMD	Ponteiro para uma estrutura de dados do tipo MQMD..
PMQMDE	Ponteiro para uma estrutura de dados do tipo MQMDE
PMQMD1	Ponteiro para uma estrutura de dados do tipo MQMD1
PMQMD2	Ponteiro para uma estrutura de dados do tipo MQMD2
PMQMHBO	Ponteiro para uma estrutura de dados do tipo MQMHBO
PMQOD	Ponteiro para uma estrutura de dados do tipo MQOD
PMQOR	Ponteiro para uma estrutura de dados do tipo MQOR
PMQPD	Ponteiro para uma estrutura de dados do tipo MQPD
PMQPID	Ponteiro para um identificador de processo MQPID
PMQPMO	Ponteiro para uma estrutura de dados do tipo MQPMO
PMQPTR	Ponteiro para dados do tipo MQPTR
PMQRFH	Ponteiro para uma estrutura de dados do tipo MQRFH
PMQRFH2	Ponteiro para uma estrutura de dados do tipo MQRFH2
PMQRMH	Ponteiro para uma estrutura de dados do tipo MQRMH
PMQRR	Ponteiro para uma estrutura de dados do tipo MQRR
PMQSCO	Ponteiro para uma estrutura de dados do tipo MQSCO
PMQSD	Ponteiro para uma estrutura de dados do tipo MQSD
PMQSMPO	Ponteiro para uma estrutura de dados do tipo MQSMPO
PMQSRO	Ponteiro para uma estrutura de dados do tipo MQSRO
PMQSTS	Ponteiro para uma estrutura de dados do tipo MQSTS
PMQTID	Ponteiro para um identificador de encadeamento MQTID
PMQTM	Ponteiro para uma estrutura de dados do tipo MQTM

Tabela 679. Tipos de dados elementares (continuação)

<b>Tipo de Dados</b>	<b>Representação</b>
PMQTM2	Ponteiro para uma estrutura de dados do tipo MQTM2
PMQUINT8	Ponteiro para dados do tipo MQUINT8
PMQUINT16	Ponteiro para dados do tipo MQUINT16
PMQUINT32	Ponteiro para dados do tipo MQUINT32
PMQUINT64	Ponteiro para dados do tipo MQUINT64
PMQULONG	Ponteiro para dados do tipo MQULONG
PMQVOID	Ponteiro
PMQWIH	Ponteiro para uma estrutura de dados do tipo MQWIH
PMQXQH	Ponteiro para uma estrutura de dados do tipo MQXQH

### IBM i **MQBOOL em IBM i**

O tipo de dados MQBOOL representa um valor booleano. O valor 0 representa false. Qualquer outro valor representa true.

Um MQBOOL deve estar alinhado como para o tipo de dados MQLONG

### IBM i **MQBYTE em IBM i**

O tipo de dados MQBYTE representa um único byte de dados

Nenhuma interpretação particular é colocada no byte—é tratada como uma sequência de bits, e não como um número binário ou caractere. Não é necessário nenhum alinhamento especial.

Uma matriz de MQBYTE às vezes é usada para representar uma área de armazenamento principal com uma natureza que não é conhecida pelo gerenciador de filas. Por exemplo, a área pode conter dados de mensagens de aplicativos ou uma estrutura. O alinhamento de limite desta área deve ser compatível com a natureza dos dados contidos dentro dela;

### IBM i **MQBYTEn (Sequência de n bytes) no IBM i**

Cada tipo de dados MQBYTEn representa uma cadeia de  $n$  bytes.

Em que  $n$  pode usar um dos seguintes valores:

- 16, 24, 32 ou 64.

Cada byte é descrito pelo tipo de dados MQBYTE. Não é necessário nenhum alinhamento especial.

Se os dados na sequência forem menores que o comprimento definido da sequência, os dados deverão ser preenchidos com nulos para preencher a sequência.

Quando o gerenciador de filas retorna sequências de bytes para o aplicativo (por exemplo, na chamada MQGET), o gerenciador de filas sempre preenche com nulos o comprimento definido da sequência.

As constantes estão disponíveis para definir os comprimentos dos campos de sequência de bytes

### IBM i **MQCHAR (caractere) em IBM i.**

O tipo de dados MQCHAR representa um único caractere..

O identificador do conjunto de caracteres codificado do caractere é aquele do gerenciador de fila (consulte o atributo **CodedCharSetId** no tópico [CodedCharSetId](#) )... Não é necessário nenhum alinhamento especial.

**Nota:** Os dados da mensagem do aplicativo especificados nas chamadas MQGET, MQPUT e MQPUT1 são descritos pelo tipo de dados MQBYTE, não pelo tipo de dados MQCHAR.

### **IBM i** **MQCHARn (Sequência de n caracteres) em IBM i**

Cada tipo de dados MQCHARn representa uma cadeia de *n* caracteres.

Em que *n* pode usar um dos seguintes valores:

- 4, 8, 12, 16, 20, 28, 32, 48, 64, 128, ou 256

Cada caractere é descrito por um tipo de dados MQCHAR. Não é necessário nenhum alinhamento especial.

Se os dados na sequência forem menores que o comprimento definido da sequência, os dados deverão ser preenchidos com espaços em branco para preencher a sequência. Em alguns casos, um caractere nulo pode ser usado para terminar a sequência prematuramente, em vez de preenchimento com espaços em branco; o caractere nulo e os caracteres que o seguem são tratados como espaços em branco, até o comprimento definido da sequência. Os locais em que um nulo pode ser usado são identificados nas descrições de chamada e de tipo de dados

Quando o gerenciador de filas retorna sequências de caracteres para o aplicativo (por exemplo, na chamada MQGET), o gerenciador de filas sempre preenche com espaços em branco para o comprimento definido da sequência; o gerenciador de filas não usa o caractere nulo para delimitar a sequência.

São disponibilizadas constantes que definem os comprimentos dos campos de sequência de caracteres

### **IBM i** **MQFLOAT32 em IBM i**

O tipo de dados MQFLOAT32 é um número de vírgula flutuante de 32 bits representado usando o formato de vírgula flutuante IEEE padrão..

Um MQFLOAT32 deve ser alinhado em um limite de 4 bytes

### **IBM i** **MQFLOAT64 em IBM i**

O tipo de dados MQFLOAT64 é um número de vírgula flutuante de 64 bits representado usando o formato de vírgula flutuante IEEE padrão

Um MQFLOAT64 deve ser alinhado em um limite de 8 bytes

### **MQHCONFIG-identificador de configuração**

O tipo de dados MQHCONFIG representa um identificador de configuração, ou seja, o componente que está sendo configurado para um serviço instalável específico Uma alça de configuração deve ser alinhada em seu limite natural

**Nota:** Os aplicativos devem testar variáveis desse tipo somente para igualdade..

### **IBM i** **MQHCONN (Identificador de Conexão) no IBM i**

O tipo de dados MQHCONN representa um identificador de conexão, ou seja, a conexão com um gerenciador de filas específico

Uma alça de conexão deve ser alinhada em seu limite natural

**Nota:** Os aplicativos devem testar variáveis desse tipo somente para igualdade..

### **IBM i** **MQHMSG (Identificador de mensagens) no IBM i**

O tipo de dados MQHMSG representa um identificador da mensagem que fornece acesso a uma mensagem

Um identificador de mensagem deve estar alinhado em um limite de 8 bytes



**Nota:** Os aplicativos devem testar variáveis desse tipo somente para igualdade..

#### **IBM i** ***MQHOBJ (Identificador de objetos) no IBM i***

O tipo de dados MQHOBJ representa uma manipulação de objetos que fornece acesso a um objeto. Uma alça de objeto deve ser alinhada em seu limite natural.

**Nota:** Os aplicativos devem testar variáveis desse tipo somente para igualdade..

#### **IBM i** ***MQINT8 (número inteiro assinado de 8 bits) em IBM i***

O tipo de dados MQINT8 é um número inteiro assinado de 8 bits que pode usar qualquer valor no intervalo de -128 a +127, a menos que seja restrito de outra forma pelo contexto.

#### **IBM i** ***MQINT16 (número inteiro assinado de 16 bits) no IBM i***

O tipo de dados MQINT16 é um número inteiro assinado de 16 bits que pode ter qualquer valor no intervalo de -32 768 a +32 767, a menos que seja restrito de outra forma pelo contexto.

Um MQINT16 deve ser alinhado em um limite de 2 bytes.

#### **IBM i** ***MQINT32 (número inteiro de 32 bits) no IBM i***

O tipo de dados MQINT32 é um número inteiro assinado de 32 bits.

Ela é equivalente a MQLONG.

#### **IBM i** ***MQINT64 (número inteiro de 64 bits) em IBM i .***

O tipo de dados MQINT64 é um número inteiro assinado de 64 bits que pode usar qualquer valor no intervalo de -9 223 372 036 854 775 808 a + 9 223 372 036 854 775 807, a menos que seja restrito pelo contexto.

Para COBOL, o intervalo válido é limitado a -999 999 999 999 999 999 até +999 999 999 999 999 999. Um MQINT64 deve ser alinhado em um limite de 8 bytes.

#### **IBM i** ***MQLONG (número inteiro longo) em IBM i***

O tipo de dados MQLONG é um número inteiro binário assinado de 32 bits que pode usar qualquer valor no intervalo -2 147 483 648 a + 2 147 483 647, a menos que seja restrito de outra forma pelo contexto, alinhado em seu limite natural.

#### ***MQPID-identificador do processo***

O identificador de processo IBM MQ .

Esse é o mesmo identificador usado em dumps de IBM MQ trace e FFST , mas pode ser diferente do identificador do processo do sistema operacional...

#### ***MQPTR-ponteiro***

O tipo de dados MQPTR é o endereço de dados de qualquer tipo.. A pointer must be aligned on its natural boundary; this is a 16-byte boundary on IBM i.

Algumas linguagens de programação suportam ponteiros digitados; o MQI também os usa em alguns casos.

#### ***MQTID-identificador de encadeamento***

O identificador de encadeamento do MQ .

Esse é o mesmo identificador usado no rastreamento do MQ e FFST dumps, mas pode ser diferente do identificador de encadeamento do sistema operacional.

### **IBM i MQUINT8 (número inteiro não assinado de 8 bits) em IBM i**

O tipo de dados MQUINT8 é um número inteiro não assinado de 8 bits que pode usar qualquer valor no intervalo de 0 a +255, a menos que seja restrito de outra forma pelo contexto..

### **MQUINT16 -Número inteiro não assinado de 16 bits**

O tipo de dados MQUINT16 é um número inteiro não assinado de 16 bits que pode ter qualquer valor no intervalo de 0 a +65 535, a menos que seja restrito de outra forma pelo contexto.

Um MQUINT16 deve ser alinhado em um limite de 2 bytes.

### **IBM i MQUINT32 (número inteiro não assinado de 32 bits) em IBM i**

O tipo de dados MQUINT32 é um inteiro não assinado de 32 bits. É equivalente a MQLONG.

### **MQUINT64 -número inteiro não assinado de 64 bits**

O tipo de dados MQUINT64 é um número inteiro não assinado de 64 bits que pode usar qualquer valor no intervalo de 0 a +18 446 744 073 709 551 615, a menos que seja restrito de outra forma pelo contexto...

Para COBOL, o intervalo válido é limitado a 0 a +999 999 999 999 999 999 999 999. Um MQUINT64 deve estar alinhado em um limite de 8 bytes

### **MQULONG-Número inteiro não assinado de 32 bits**

O tipo de dados MQULONG é um número inteiro binário não assinado de 32 bits que pode usar qualquer valor no intervalo de 0 a + 4 294 967 294, a menos que seja restrito de outra forma pelo contexto...

Um MQULONG deve ser alinhado em um limite de 4 bytes

### **PMQACH-ponteiro para uma estrutura de dados do tipo MQACH;**

Um ponteiro para uma estrutura de dados do tipo MQACH

### **PMQAIR-ponteiro para uma estrutura de dados do tipo MQAIR**

Um ponteiro para uma estrutura de dados do tipo MQAIR

### **Ponteiro PMQAXC para uma estrutura de dados do tipo MQAXC**

Um ponteiro para uma estrutura de dados do tipo MQAXC

### **Ponteiro PMQAXP para uma estrutura de dados do tipo MQAXP**

Um ponteiro para uma estrutura de dados do tipo MQAXP

### **PMQBMHO-ponteiro para uma estrutura de dados do tipo MQBMHO**

Um ponteiro para uma estrutura de dados do tipo MQBMHO

### **PMQBO-ponteiro para uma estrutura de dados do tipo MQBO**

Um ponteiro para uma estrutura de dados do tipo MQBO

***PMQBOOL-ponteiro para dados do tipo MQBOOL***

Um ponteiro para os dados do tipo MQBOOL

Um ponteiro para os dados do tipo MQBOOL

***PMQBYTE-ponteiro para um tipo de dados de MQBYTE***

Um ponteiro para um tipo de dados de MQBYTE

***Ponteiro PMQBYTEn para uma estrutura de dados do tipo MQBYTEn***

Um ponteiro para uma estrutura de dados do tipo MQBYTEn, em que n pode ser 8, 12, 16, 24, 32, 40, 48 ou 128.

***PMQCBC-ponteiro para uma estrutura de dados do tipo MQCBC***

Um ponteiro para uma estrutura de dados do tipo MQCBC.

***Ponteiro PMQCBD para uma estrutura de dados do tipo MQCBD***

Um ponteiro para uma estrutura de dados do tipo MQCBD

***PMQCHAR-ponteiro para dados do tipo MQCHAR***

Um ponteiro para os dados do tipo MQCHAR

***Ponteiro PMQCHARV para uma estrutura de dados do tipo MQCHARV***

Um ponteiro para uma estrutura de dados do tipo MQCHARV

***PMQCHARn-ponteiro para um tipo de dados de MQCHARn***

Um ponteiro para um tipo de dados de MQCHARn, em que n pode ser 4, 8, 12, 20, 28, 32, 64, 256, 264.

***PMQCIH-ponteiro para uma estrutura de dados do tipo MQCIH***

Um ponteiro para uma estrutura de dados do tipo de MQCIH

***PMQCMHO-ponteiro para uma estrutura de dados do tipo MQCMHO.***

Um ponteiro para uma estrutura de dados do tipo MQCMHO

***PMQCNO-ponteiro para uma estrutura de dados do tipo MQCNO***

Um ponteiro para uma estrutura de dados do tipo de MQCNO

***PMQCSP-ponteiro para uma estrutura de dados do tipo MQCSP***

Um ponteiro para uma estrutura de dados do tipo MQCSP

***PMQCTLO-ponteiro para uma estrutura de dados do tipo MQCTLO***

Um ponteiro para uma estrutura de dados do tipo MQCTLO

***Ponteiro PMQDH para uma estrutura de dados do tipo MQDH***

Um ponteiro para uma estrutura de dados do tipo MQDH

***PMQDHO-ponteiro para uma estrutura de dados do tipo MQDHO***

Um ponteiro para uma estrutura de dados do tipo MQDHO

***PMQDLH-ponteiro para uma estrutura de dados do tipo MQDLH***

Um ponteiro para uma estrutura de dados do tipo de MQDLH

***PMQDMHO-ponteiro para uma estrutura de dados de tipo MQDMHO***

Um ponteiro para uma estrutura de dados do tipo MQDMHO

***PMQDMPO-ponteiro para uma estrutura de dados do tipo MQDMPO***

Um ponteiro para uma estrutura de dados do tipo MQDMPO

Um ponteiro para uma estrutura de dados do tipo MQDMPO

***Ponteiro de PMQEPH para uma estrutura de dados do tipo MQEPH***

Um ponteiro para uma estrutura de dados do tipo MQEPH

***PMQFLOAT32 -ponteiro para dados do tipo MQFLOAT32***

Um ponteiro para os dados do tipo MQFLOAT32

***PMQFLOAT64 -ponteiro para dados do tipo MQFLOAT64***

Um ponteiro para os dados do tipo MQFLOAT64

***PMQFUNC-ponteiro para uma função***

Um ponteiro para uma função.

***Ponteiro PMQGMO para uma estrutura de dados do tipo MQGMO***

Um ponteiro para uma estrutura de dados do tipo MQGMO

***PMQHCONFIG-ponteiro para um tipo de dados de MQHCONFIG.***

Um ponteiro para um tipo de dados de MQHCONFIG

***PMQHCONN-ponteiro para um tipo de dados de MQHCONN***

Um ponteiro para um tipo de dados MQHCONN.

***PMQHMSG-ponteiro para um tipo de dados de MQHMSG***

Um ponteiro para um tipo de dados de MQHMSG

***PMQHOBJ-ponteiro para dados do tipo MQHOBJ***

Um ponteiro para os dados do tipo MQSMPO

***Ponteiro PMQIIH para uma estrutura de dados do tipo MQIIH***

Um ponteiro para uma estrutura de dados do tipo MQIIH

***PMQIMPO-ponteiro para uma estrutura de dado do tipo MQIMPO***

Um ponteiro para uma estrutura de dados do tipo MQIMPO

***PMQINT8 -ponteiro para dados do tipo MQINT8***

Um ponteiro para os dados do tipo MQINT8

***PMQINT16 -ponteiro para dados do tipo MQINT16***

Um ponteiro para os dados do tipo MQINT16

***IBM i PMQINT32 (Ponteiro para dados do tipo MQINT32) no IBM i***

O tipo de dados PMQINT32 é um ponteiro para dados do tipo MQINT32.. Ele é equivalente a PMQLONG

***IBM i PMQINT64 (Ponteiro para dados do tipo MQINT64) em IBM i***

O tipo de dados PMQINT64 é um ponteiro para dados do tipo MQINT64

***PMQLONG-ponteiro para dados do tipo MQLONG***

Um ponteiro para os dados do tipo MQLONG

***PMQMD-ponteiro para estrutura do tipo MQMD***

Um ponteiro para a estrutura do tipo MQMD

***PMQMDE-ponteiro para uma estrutura de dados do tipo MQMDE***

Um ponteiro para uma estrutura de dados do tipo MQMDE

***PMQMDI-ponteiro para uma estrutura de dados do tipo MQMDI;***

Um ponteiro para uma estrutura de dados do tipo MQMDI

***PMQMD2 -ponteiro para uma estrutura de dados do tipo MQMD2***

Um ponteiro para uma estrutura de dados do tipo MQMD2

***PMQMHBO-ponteiro para uma estrutura de dados do tipo MQMHBO***

Um ponteiro para uma estrutura de dados do tipo MQMHBO..

***Ponteiro PMQOD para uma estrutura de dados do tipo MQOD***

Um ponteiro para uma estrutura de dados do tipo MQOD

***PMQOR-ponteiro para uma estrutura de dados do tipo MQOR;***

Um ponteiro para uma estrutura de dados do tipo MQOR

***Ponteiro PMQPD para uma estrutura de dados do tipo MQPD***

Um ponteiro para uma estrutura de dados do tipo MQPD

***PMQPID-ponteiro para um identificador de processo***

Um ponteiro para um identificador de processo.

***Ponteiro PMQPMO para uma estrutura de dados do tipo MQPMO***

Um ponteiro para uma estrutura de dados do tipo MQPMO

***PMQPTR-ponteiro para dados do tipo MQPTR***

Um ponteiro para os dados do tipo MQPTR

***PMQRFH-ponteiro para uma estrutura de dados do tipo MQRFH;***

Um ponteiro para uma estrutura de dados do tipo MQRFH

***PMQRFH2 -ponteiro para uma estrutura de dados do tipo MQRFH2***

Um ponteiro para uma estrutura de dados do tipo MQRFH2.

***PMQRMH-ponteiro para uma estrutura de dados do tipo MQRMH***

Um ponteiro para uma estrutura de dados do tipo MQRMH

***PMQRR-ponteiro para uma estrutura de dados do tipo MQRR.***

Um ponteiro para uma estrutura de dados do tipo MQRR

***Ponteiro PMQSCO para uma estrutura de dados do tipo MQSCO***

Um ponteiro para uma estrutura de dados do tipo MQSCO.

**Ponteiro *PMQSD* para uma estrutura de dados do tipo *MQSD***

Um ponteiro para uma estrutura de dados do tipo *MQSD*

***PMQSMPO*-ponteiro para uma estrutura de dados do tipo *MQSMPO***

Um ponteiro para uma estrutura de dados do tipo *MQSMPO*

**Ponteiro *PMQSRO* para uma estrutura de dados do tipo *MQSRO***

Um ponteiro para uma estrutura de dados do tipo *MQSRO*

***PMQSTS*-ponteiro para uma estrutura de dados do tipo *MQSTS***

Um ponteiro para uma estrutura de dados do tipo *MQSTS*

**Ponteiro *PMQTID* para uma estrutura de dados do tipo *MQTID***

Um ponteiro para uma estrutura de dados do tipo *MQTID*.

***PMQTM*-ponteiro para uma estrutura de dados do tipo *MQTM***

Um ponteiro para uma estrutura de dados do tipo *MQTM*

***PMQPMC2* -ponteiro para uma estrutura de dados do tipo *MQPMC2* .**

Um ponteiro para uma estrutura de dados do tipo *MQPMC2*.

***PMQUINT8* -ponteiro para dados do tipo *MQUINT8***

Um ponteiro para os dados do tipo *MQUINT8*

***PMQUINT16* -ponteiro para dados do tipo *MQUINT16***

Um ponteiro para os dados do tipo *MQUINT16*

**IBM i *PMQUINT32* (Ponteiro para dados do tipo *MQUINT32*) em IBM i**

O tipo de dados *PMQUINT32* é um indicador para dados do tipo *MQUINT32*. É equivalente a *PMQLONG*.

**IBM i *PMQUINT64* (Ponteiro para dados do tipo *MQUINT64*) no IBM i**

O tipo de dados *PMQUINT64* é um indicador para dados do tipo *MQUINT64*.

***PMQLONG*-ponteiro para dados do tipo *MQLONG***

Um ponteiro para os dados do tipo *MQLONG*

***PMQVOID*-ponteiro**

Um ponteiro. "

## **Ponteiro PMQWIH para uma estrutura de dados do tipo MQWIH**

Um ponteiro para uma estrutura de dados do tipo MQWIH

## **PMQXQH-ponteiro para uma estrutura de dado do tipo MQXQH**

Um ponteiro para uma estrutura de dados do tipo MQXQH

## **Considerações sobre o idioma**

Este tópico contém informações para ajudá-lo a usar o MQI da linguagem de programação RPG.

Algumas dessas considerações de linguagem são:

- [“arquivos de cópia”](#) na página 1044
- [“Chamadas”](#) na página 1046
- [“parâmetros de chamada”](#) na página 1046
- [“Estruturas”](#) na página 1046
- [“Constantes nomeadas”](#) na página 1047
- [“procedimentos de MQI”](#) na página 1047
- [“Considerações de encadeamento”](#) na página 1047
- [“Controle de consolidações”](#) na página 1048
- [“Codificando as chamadas ligadas”](#) na página 1048
- [“Convenções de notação”](#) na página 1049

## **arquivos de cópia**

Vários arquivos COPY são fornecidos para ajudar na gravação de programas de aplicativos RPG que usam enfileiramento de mensagem. Há três conjuntos de arquivos COPY:

- Arquivos COPY com nomes que terminam com a letra *G* são para uso com programas que usam ligação estática. Esses arquivos são inicializados com as exceções indicadas em [“Estruturas”](#) na página 1046
- Arquivos COPY com nomes terminados com a letra *H* são para uso com programas que usam ligação estática, mas **não** são inicializados.
- Arquivos COPY com nomes que terminam com a letra *R* são para uso com programas que usam ligação dinâmica. Esses arquivos são inicializados com as exceções indicadas em [“Estruturas”](#) na página 1046

Os arquivos COPY residem em QRPGLSRC na biblioteca QMQM..

Para cada conjunto de arquivos COPY, há dois arquivos que contêm constantes nomeadas e um arquivo para cada uma das estruturas. Os arquivos COPY são resumidos em [Tabela 680](#) na página 1044

<b>Nome do arquivo (ligação estática, inicializado, CMQ* G)</b>	<b>Nome do arquivo (ligação estática, não inicializado, CMQ* H)</b>	<b>Nome do arquivo (ligação dinâmica, inicializado, CMQ* R)</b>	<b>Índice</b>
CMQBOG	CMQBOH	-	Iniciar estrutura de opções
CMQCDG	CMQCDH	CMQCDR	Estrutura de definição de canal
CMQCFBG	CMQCFBH	-	parâmetro de filtro de bits PCF
CMQCFG	-	-	Constantes para PCF e eventos



Tabela 680. Arquivos RPG COPY (continuação)

<b>Nome do arquivo (ligação estática, inicializado, CMQ* G)</b>	<b>Nome do arquivo (ligação estática, não inicializado, CMQ* H)</b>	<b>Nome do arquivo (ligação dinâmica, inicializado, CMQ* R)</b>	<b>Índice</b>
CMQCFBSG	CMQCFBSH	-	Sequência de bytes PCF
CMQCFGRG	CMQCFGRH	-	parâmetro do grupo PCF
CMQCFIFG	CMQCFIFH	-	Parâmetro de filtro de número inteiro PCF
CMQCFHG	CMQCFHH	-	Cabeçalho PCF
CMQCFILG	CMQCFILH	-	Estrutura do parâmetro da lista de inteiros PCF
CMQCFING	CMQCFINH	-	Estrutura de parâmetro de número inteiro PCF
CMQCFSG	CMQCFSH	-	Parâmetro de filtro de sequência PCF
CMQCFSLG	CMQCFSLH	-	Estrutura do parâmetro da lista de sequências PCF.
CMQCFSTG	CMQCFSTH	-	Estrutura do parâmetro de sequência PCF
CMQCFXLG	CMQCFXLH	-	Nome abreviado PCF para CFIL64
CMQCFXNG	CMQCFXNH	-	Nome abreviado PCF para CFIN64
CMQCIHG	CMQCIHH	-	Estrutura do cabeçalho de informações CICS
CMQCNQG	CMQCNQH	-	Conectar estrutura de opções
CMQCSPG	CMQCSPH	-	Parâmetros de segurança
CMQCXPG	CMQCXPH	CMQCXPR	Estrutura do parâmetro de saída do canal
CMQDHG	CMQDHH	CMQDHR	Estrutura do cabeçalho de distribuição
CMQDLHG	CMQDLHH	CMQDLHR	Estrutura do cabeçalho de devoluções
CMQDXPG	CMQDXPH	CMQDXPR	Estrutura do parâmetro de saída de conversão de dados
CMQEPHG	CMQEPHH	-	Estrutura do cabeçalho PCF integrado
CMQG	-	CMQR	Constantes nomeadas para MQI principal
CMQGMQG	CMQGMQH	CMQGMOR	Obter estrutura de opções de mensagem
CMQIIHG	CMQIIHH	CMQIIHR	Estrutura do cabeçalho de informações IMS
CMQMDEG	CMQMDEH	CMQMDER	Estrutura de extensão do descritor de mensagens
CMQMDG	CMQMDH	CMQMDR	Estrutura do descritor de mensagem

Tabela 680. Arquivos RPG COPY (continuação)

Nome do arquivo (ligação estática, inicializado, CMQ* G)	Nome do arquivo (ligação estática, não inicializado, CMQ* H)	Nome do arquivo (ligação dinâmica, inicializado, CMQ* R)	Índice
CMQMD1G	CMQMD1H	CMQMD1R	Estrutura do descritor de mensagens versão 1
CMQMD2G	CMQMD2H	-	Estrutura do descritor de mensagens versão 2
CMQODG	CMQODH	CMQODR	Estrutura do descritor de objeto
CMQORG	CMQORH	CMQORR	Estrutura do registro de objeto
CMQPMOG	CMQPMOH	CMQPMOR	Estrutura de opções de mensagem put
CMQPSG	-	-	Constantes para publicação / assinatura
CMQRFHG	CMQRFHH	-	Estrutura do cabeçalho de regras e formatação
CMQRFH2G	CMQRFH2H	-	Estrutura do cabeçalho 2 de regras e formatação
CMQRMHG	CMQRMHH	CMQRMHR	Estrutura do cabeçalho da mensagem de referência
CMQRRG	CMQRRH	CMQRRR	Estrutura do registro de resposta:
CMQTMCG	CMQTMCH	CMQTMCR	Estrutura da mensagem do acionador (formato de caractere).
CMQTM2G	CMQTM2H	CMQTM2R	Estrutura da mensagem acionadora (formato de caractere) versão 2
CMQTMG	CMQTMH	CMQTMR	Estrutura da mensagem do acionador
CMQWIHG	CMQWIHH	-	Estrutura do cabeçalho de informações de trabalho
CMQXG	-	CMQXR	Constantes nomeadas para saída de conversão de dados
CMQXQHG	CMQXQHH	CMQXQHR	Estrutura do cabeçalho da fila de transmissão

## Chamadas

Chamadas são descritas usando seus nomes individuais.

## parâmetros de chamada

Alguns parâmetros passados para o MQI podem ter mais de uma função simultânea. Isso ocorre porque o valor de número inteiro passado geralmente é testado na configuração de bits individuais dentro do campo e não em seu valor total. Isso permite 'incluir' várias funções juntas e passá-las como um único parâmetro.

## Estruturas

Todas as estruturas IBM MQ são definidas com valores iniciais para os campos, com as seguintes exceções:

- Qualquer estrutura com um sufixo de H.
- MQTMC
- MQTMC2

Estes valores iniciais são definidos na tabela relevante para cada estrutura.

As declarações de estrutura não contêm instruções DS .. Isso permite que o aplicativo declare uma única estrutura de dados ou uma estrutura de dados de várias ocorrências, codificando a instrução DS e, em seguida usando a instrução /COPY para copiar no restante da declaração:

```
D*..1.....2.....3.....4.....5.....6.....7
D* Declare an MQMD data structure with 5 occurrences
DMYMD      DS              5
D/COPY CMQMDR
```

## Constantes nomeadas

Há muitos valores de número inteiro e de caracteres que fornecem troca de dados entre seu programa de aplicativo e o gerenciador de filas. Para facilitar uma abordagem mais legível e consistente para usar esses valores, constantes nomeadas são definidas para eles. É possível usar essas constantes nomeadas e não os valores que elas representam, pois isso melhora a capacidade de leitura do código-fonte do programa

Quando o arquivo COPY CMQG é incluído em um programa para definir as constantes, o compilador RPG emitirá muitas mensagens de gravidade zero para as constantes que não são usadas pelo programa; essas mensagens são benignas e podem ser ignoradas com segurança.

## procedimentos de MQI

Ao usar as chamadas ligadas de ILE, deve-se ligar aos procedimentos MQI ao criar seu programa. Esses procedimentos são exportados dos seguintes programas de serviços, conforme apropriado:

### QMQM/LIBMQM

Este programa de serviços contém as ligações de encadeamento único para a versão 5.1 e superior. Consulte a seção a seguir para obter considerações especiais ao gravar aplicativos encadeados.

### QMQM/LIBMQM\_R

Este programa de serviços contém as ligações multiencadeadas para a versão 5.1 e superior. Consulte a seção a seguir para obter considerações especiais ao gravar aplicativos encadeados.

### QMQM/LIBMQIC

Este programa de serviços é para ligar aplicativos clientes não encadeados.

### QMQM/LIBMQIC\_R

Este programa de serviços é para aplicativos clientes encadeados de ligação.

Use o comando CRTPGM para criar seus programas. Por exemplo, o comando a seguir cria um programa de encadeamento único que usa as chamadas de limite ILE:

```
CRTPGM PGM(MYPROGRAM) BNDSRVPGM(QMQM/LIBMQM)
```

## Considerações de encadeamento

O compilador RPG usado para IBM i faz parte do WebSphere Development Toolset e do WebSphere Development Studio for IBM i e é conhecido como o ILE RPG IV Compiler.

Em geral, os programas de RPG não devem usar os programas de serviços multiencadeados. Exceções são programas RPG criados usando o compilador ILE RPG IV e que contêm a palavra-chave THREAD(\*SERIALIZE) na especificação de controle. No entanto, mesmo que esses programas sejam thread-safe, deve-se considerar cuidadosamente o design geral do aplicativo, pois o

THREAD(\*SERIALIZE) força a serialização de procedimentos RPG no nível do módulo e isso pode ter um efeito adverso no desempenho geral.

Quando os programas RPG são usados como saídas de conversão de dados, eles devem se tornar thread-safe e devem ser recompilados usando o compilador ILE RPG da versão 4.4 ou superior, com THREAD(\*SERIALIZE) especificado na especificação de controle

Para obter informações adicionais sobre encadeamento, consulte o *IBM i IBM MQ Development Studio: ILE RPG Reference* e o *IBM i IBM MQ Development Studio: ILE RPG Programmer's Guide*.

## Controle de consolidações

As funções de ponto de sincronização do MQI MQCMIT e MQBACK estão disponíveis para programas ILE RPG em execução no modo normal; essas chamadas permitem que o programa confirme e retorne as mudanças para os recursos do MQ

## Codificando as chamadas ligadas

Os procedimentos ILE MQI são listados em [Tabela 681](#) na página 1048.

<i>Tabela 681. Chamadas ligadas ILE RPG suportadas por cada programa de serviços</i>		
<b>Nome da chamada</b>	<b>LIBMQM e LIBMQM_R</b>	<b>LIBMQIC e LIBMQIC_R</b>
MQBACK	Y	Y
MQBEGIN	Y	Y
MQCMIT	Y	Y
MQCLOSE	Y	Y
MQCONN	Y	Y
MQCONNX	Y	Y
MQDISC	Y	Y
MQGET	Y	Y
MQINQ	Y	Y
MQOPEN	Y	Y
MQPUT	Y	Y
MQPUT1	Y	Y
MQSET	Y	Y
MQXCNVC	Y	Y

Para usar esses procedimentos, você precisa:

1. Defina os procedimentos externos em suas especificações D'. Todos eles estão disponíveis no CMQG do membro do arquivo COPY contendo as constantes nomeadas.
2. Use o código de operação CALLP para chamar o procedimento juntamente com seus parâmetros

Por exemplo, a chamada MQOPEN requer a inclusão do seguinte código:

```
D*****
D**  MQOPEN Call -- Open Object (From COPY file CMQG)      **
D*****
D*
D*. .1....:....2....:....3....:....4....:....5....:....6....:....7..
DMQOPEN          PR          EXTPROC('MQOPEN')
D* Connection handle
```

```

D HCONN                10I 0 VALUE
D* Object descriptor
D OBJDSC                224A
D* Options that control the action of MQOPEN
D OPTS                 10I 0 VALUE
D* Object handle
D HOBJ                 10I 0
D* Completion code
D CMPCOD               10I 0
D* Reason code qualifying CMPCOD
D REASON               10I 0
D*

```

Para chamar o procedimento, depois de inicializar os vários parâmetros, você precisa do seguinte código:

```

...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ...+... 8
  C                CALLP      MQOPEN(HCONN : MQOD : OPTS : HOBJ :
  C                CMPCOD : REASON)

```

Aqui, a estrutura MQOD é definida usando o membro COPY CMQODG que o divide em seus componentes..

## Convenções de notação

Os últimos tópicos nesta seção mostram como:

- As chamadas devem ser chamadas
- Os parâmetros devem ser declarados.
- Vários tipos de dados devem ser declarados

Em vários casos, os parâmetros são matrizes ou sequências de caracteres com um tamanho que não é fixo. Para estes, um "n" minúsculo é usado para representar uma constante numérica.. Quando a declaração para esse parâmetro for codificada, o "n" deverá ser substituído pelo valor numérico necessário.

## MQAIR (Registro de informações sobre autenticação) no IBM i

A estrutura MQAIR representa o registro de informações sobre autenticação..

### Visão Geral

**Propósito:** A estrutura MQAIR permite que um aplicativo em execução como um cliente IBM MQ especifique informações sobre um autenticador que deve ser usado para a conexão do cliente. A estrutura é um parâmetro de saída na chamada MQCONN.

**Conjunto de caracteres e codificação:** os dados em MQAIR devem estar no conjunto de caracteres fornecido pelo atributo do gerenciador de fila **CodedCharSetId** e a codificação do gerenciador de filas local fornecido por ENNAT.

- [“Campos” na página 1049](#)
- [“Valores iniciais” na página 1051](#)
- [“Declaração RPG” na página 1052](#)

### Campos

A estrutura MQAIR contém os campos a seguir; os campos são descritos em **ordem alfabética**:

#### AICN (número inteiro assinado de 10 dígitos)

Este é o nome do host ou o endereço de rede de um host no qual o servidor LDAP está em execução Este pode ser seguido por um número de porta opcional, colocado entre parênteses.

Se o valor for menor que o comprimento do campo, termine o valor com um caractere nulo ou preencha-o com espaços em branco até o comprimento do campo. Se o valor não for válido, a chamada falhará com o código de razão RC2387.

O número da porta padrão é 389.

Este é um campo de entrada. O comprimento deste campo é fornecido pelo LNAICN O valor inicial deste campo é caracteres em branco.

#### **AITYP (número inteiro assinado de 10 dígitos)**

Este é o tipo de informações sobre autenticação contidas no registro

O valor deve ser:.

#### **AITLDP**

Revogação de certificado usando servidor LDAP.

Se o valor não for válido, a chamada falhará com código de razão RC2386.

Este é um campo de entrada. O valor inicial deste campo é AITLDP.

#### **AIPW (número inteiro assinado de 10 dígitos)**

Essa é a senha necessária para acessar o servidor de CRL LDAP

Se o valor for menor que o comprimento do campo, termine o valor com um caractere nulo ou preencha-o com espaços em branco até o comprimento do campo. Se o servidor LDAP não requerer uma senha ou você omitir o nome do usuário LDAP, *AIPW* deverá ser nulo ou em branco. Se você omitir o nome do usuário LDAP e *AIPW* não for nulo ou em branco, a chamada falhará com o código de razão RC2390.

Este é um campo de entrada. O comprimento deste campo é fornecido por LNLDPW O valor inicial desse campo em branco.

#### **AILUL (número inteiro assinado de 10 dígitos)**

Esse é o comprimento em bytes do nome do usuário LDAP endereçado pelo campo *AILUP* ou *AILUO*. O valor deve estar no intervalo zero a LNDISN. Se o valor não for válido, a chamada falha com o código de razão RC2389.

Se o servidor LDAP envolvido não requerer um nome de usuário, configure esse campo como zero.

Este é um campo de entrada. O valor inicial desse campo é 0.

#### **AILUO (número inteiro assinado de 10 dígitos)**

Este é o deslocamento em bytes do nome do usuário LDAP do início da estrutura MQAIR.

O deslocamento pode ser positivo ou negativo O campo será ignorado se *LDAPUserNameLength* for zero.

É possível usar um *LDAPUserNamePtr* ou *LDAPUserNameOffset* para especificar o nome do usuário LDAP, mas não ambos; consulte a descrição do campo *LDAPUserNamePtr* para obter detalhes

Este é um campo de entrada. O valor inicial desse campo é 0.

#### **AILUP (número inteiro assinado de 10 dígitos)**

Este é o nome do usuário do LDAP

Ele consiste no Nome Distinto do usuário que está tentando acessar o servidor CRL do LDAP Se o valor for menor que o comprimento especificado por *AILUL*, finalize o valor com um caractere nulo ou preencha com espaços em branco para o comprimento *AILUL*. O campo será ignorado se *AILUL* for zero.

É possível fornecer o nome do usuário LDAP de uma das duas maneiras:

- Usando o campo de ponteiro *AILUP*.

Nesse caso, o aplicativo pode declarar uma sequência separada da estrutura MQAIR e configurar *AILUP* para o endereço da sequência.

Considere o uso do *AILUP* para linguagens de programação que suportam o tipo de dados do ponteiro de uma maneira que seja móvel para ambientes diferentes (por exemplo, a linguagem de programação C)

- Usando o campo de deslocamento *AILUO*

Nesse caso, o aplicativo deve declarar uma estrutura composta contendo a estrutura *MQSCO* seguida pela matriz de registros *MQAIR* seguidos pelas sequências de nome do usuário LDAP e configurar *AILUO* para o deslocamento da sequência de nome apropriada do início da estrutura *MQAIR*. Assegure-se de que esse valor esteja correto e tenha um valor que possa ser acomodado em um *MQLONG* (a linguagem de programação mais restritiva é COBOL, para o qual o intervalo válido é -999 999 999 a +999 999 999).

Considere o uso do *AILUO* para linguagens de programação que não suportam o tipo de dados do ponteiro ou que implementam o tipo de dados do ponteiro de uma maneira que pode não ser móvel para ambientes diferentes (por exemplo, a linguagem de programação COBOL)..

Qualquer técnica escolhida, use apenas um de *AILUP* e *AILUO* ; a chamada falha com código de razão RC2388.

Este é um campo de entrada. O valor inicial deste campo é o ponteiro nulo nas linguagens de programação que suportam ponteiros e, caso contrário, uma cadeia de bytes all-null.

**Nota:** Nas plataformas em que a linguagem de programação não suporta o tipo de dados do ponteiro, este campo é declarado como uma cadeia de bytes do comprimento apropriado.

#### **AISID (número inteiro assinado de 10 dígitos)**

O valor deve ser:.

##### **AISIDV**

Identificador para o registro de informações sobre autenticação.

Isto é sempre um campo de entrada. O valor inicial desse campo é AISIDV.

#### **AIVER (número inteiro assinado de 10 dígitos)**

O valor deve ser:.

##### **AIVER1**

Registro de informações sobre autenticação Version-1 .

A constante a seguir especifica o número da versão atual:

##### **AIRVERC**

A versão atual do registro de informações sobre autenticação.

Isto é sempre um campo de entrada. O valor inicial desse campo é AIVER1.

### **Valores iniciais**

<i>Tabela 682. Campos em MQAIR para MQAIR.</i>		
<b>Nome de Campo</b>	<b>Nome da constante</b>	<b>Valor da constante</b>
<i>AISID</i>	AISIDV	'AIR→'
<i>AIVER</i>	AIVERC	1
<i>AITYP</i>	AITLDP	1
<i>AICN</i>	Nenhum	Cadeia nula ou espaços em branco
<i>AILUP</i>	Nenhum	Ponteiro nulo ou bytes nulos
<i>AILUO</i>	Nenhum	0

Tabela 682. Campos em MQAIR para MQAIR. (continuação)

Nome de Campo	Nome da constante	Valor da constante
AILUL	Nenhum	0
AIPW	Nenhum	Cadeia nula ou espaços em branco

**Notas:**

1. O símbolo – representa um único caractere em branco.

**Declaração RPG**

```

D*..1.....2.....3.....4.....5.....6.....7..
D* MQAIR Structure
D*
D* Structure identifier
D AISID          1      4    INZ('AIR ')
D* Structure version number
D AIVER          5      8I 0  INZ(1)
D* Type of authentication information
D AITYP          9     12I 0  INZ(1)
D* Connection name of CRL LDAP server
D AICN          13     276    INZ
D* Address of LDAP user name
D AILUP         277     292*   INZ(*NULL)
D* Offset of LDAP user name from start of MQAIR structure
D AILUO         293     296I 0  INZ(0)
D* Length of LDAP user name
D AILUL         297     300I 0  INZ(0)
D* Password to access LDAP server
D AIPW         301     332    INZ
    
```

**IBM i MQBMHO (Buffer para opções de manipulação de mensagens) no IBM i**

Estrutura que define o buffer para as opções do identificador de mensagens

**Visão Geral**

**Propósito:** A estrutura MQBMHO permite que aplicativos especifiquem opções que controlam como os identificadores de mensagens são produzidos a partir de buffers. A estrutura é um parâmetro de entrada na chamada MQBUFMH..

**Conjunto de caracteres e codificação:** os dados em MQBMHO devem estar no conjunto de caracteres do aplicativo e codificação do aplicativo (ENNAT).

- [“Campos” na página 1052](#)
- [“Valores iniciais” na página 1053](#)
- [“Declaração RPG” na página 1053](#)

**Campos**

A estrutura MQBMHO contém os seguintes campos; os campos são descritos em **ordem alfabética**:

**BMSID (número inteiro assinado de 10 dígitos).**

Buffer para estrutura de manipulação de mensagem-campo StructId

Este é o identificador de estrutura. O valor deve ser:.

**BMSIDV**

Identificador para a estrutura de manipulação de buffer para mensagem.



Isto é sempre um campo de entrada. O valor inicial desse campo é BMSIDV.

### **BMVER (número inteiro assinado de 10 dígitos).**

Buffer para a estrutura do identificador de mensagens-campo Versão

Este é o número da versão da estrutura O valor deve ser:.

#### **BMVER1**

Número da versão do buffer para a estrutura do identificador de mensagens

A constante a seguir especifica o número da versão atual:

#### **BMVERVC**

Versão atual da estrutura de buffer para manipulação de mensagens.

Isto é sempre um campo de entrada. O valor inicial desse campo é BMVER1.

### **BMOPT (número inteiro assinado de 10 dígitos)**

Buffer para a estrutura do identificador de mensagens-campo Opções

O valor do pode ser:

#### **BMDLPR**

As propriedades incluídas na manipulação de mensagens são excluídas do buffer. Se a chamada falhar nenhuma propriedade será excluída.

Opções padrão: Se você não precisar da opção descrita, use a seguinte opção:

#### **BMNONE**

Nenhuma opção especificada.

Isto é sempre um campo de entrada. O valor inicial desse campo é BMDLPR.

## **Valores iniciais**

<i>Tabela 683. Campos em MQBMHO</i>		
<b>Nome de Campo</b>	<b>Nome da constante</b>	<b>Valor da constante</b>
<i>BMSID</i>	BMSIDV	'BMHO'
<i>BMVER</i>	BMVER1	1
<i>BMOPT</i>	BMNONE	0

## **Declaração RPG**

```
D* MQBMHO Structure
D*
D*
D* Structure identifier
D BMSID          1      4   INZ('BMHO')
D*
D* Structure version number
D BMVER          5      8I 0 INZ(1)
D*
D* Options that control the action of MQBUFMH
D BMOPT          9      12I 0 INZ(1)
```

## **IBM i MQBO (Iniciar opções) em IBM i**

A estrutura MQBO permite que o aplicativo especifique opções relacionadas à criação de uma unidade de trabalho..

## Visão Geral

**Propósito:** A estrutura é um parâmetro de entrada / saída na chamada MQBEGIN.

**Conjunto de caracteres e codificação:** os dados em MQBO devem estar no conjunto de caracteres fornecido pelo atributo do gerenciador de filas **CodedCharSetId** e a codificação do gerenciador de fila local fornecido por ENNAT.

- “Campos” na página 1054
- “Valores iniciais” na página 1054
- “Declaração RPG” na página 1055

## Campos

A estrutura MQBO contém os campos a seguir; os campos são descritos em **ordem alfabética**:

### **BOOPT (número inteiro assinado de 10 dígitos)**

As opções que controlam a ação de MQBEGIN

O valor deve ser:.

#### **BONONE**

Nenhuma opção especificada.

Isto é sempre um campo de entrada. O valor inicial deste campo é BONONE

### **BOSID (sequência de caracteres de 4 bytes).**

Identificador de estruturação.

O valor deve ser:.

#### **BOSIDV**

Identificador para a estrutura de opções de início

Isto é sempre um campo de entrada. O valor inicial desse campo é BOSIDV.

### **BOVER (número inteiro assinado de 10 dígitos)**

Número de versão da estrutura.

O valor deve ser:.

#### **BOVER1**

Número da versão para a estrutura de opções de início

A constante a seguir especifica o número da versão atual:

#### **BOVERC**

Versão atual da estrutura de opções iniciais.

Isto é sempre um campo de entrada. O valor inicial desse campo é BOVER1.

## Valores iniciais

Nome de Campo	Nome da constante	Valor da constante
<i>BOSID</i>	BOSIDV	'B0--'
<i>BOVER</i>	BOVER1	1
<i>BOOPT</i>	BONONE	0

### Notas:

1. O símbolo – representa um único caractere em branco.

## Declaração RPG

```
D*..1....:....2....:....3....:....4....:....5....:....6....:....7..
D* MQBO Structure
D*
D* Structure identifier
D BOSID          1          4    INZ('BO ')
D* Structure version number
D BOVER          5          8I 0 INZ(1)
D* Options that control the action of MQBEGIN
D BOOPT          9          12I 0 INZ(0)
```

## IBM i MQCBC (contexto de retorno de chamada) em IBM i

A estrutura que descreve a rotina de retorno de chamada

### Visão Geral

#### Propósito

A estrutura MQCBC é usada para especificar informações de contexto que são transmitidas para uma função de retorno de chamada.

A estrutura é um parâmetro de entrada / saída na chamada para uma rotina do consumidor de mensagens

#### Versão

A versão atual do MQCBC é CBCV2.

#### Conjunto de caracteres e codificação

Os dados no MQCBC estão no conjunto de caracteres fornecido pelo atributo do gerenciador de fila do **CodedCharSetId** e pela codificação do gerenciador de fila local fornecido pelo ENNAT. No entanto, se o aplicativo estiver em execução como um cliente IBM MQ, a estrutura estará no conjunto de caracteres e na codificação do cliente..

- [“Campos” na página 1055](#)
- [“Valores iniciais” na página 1061](#)
- [“Declaração RPG” na página 1061](#)

### Campos

A estrutura MQCBC contém os campos a seguir; os campos são descritos em ordem alfabética:

#### CBCBUFFLEN (número inteiro assinado de 10 dígitos)

O buffer pode ser maior que o valor de Comprimento MaxMsgdefinido para o consumidor e o valor ReturnedLength no MQGMO.

Estrutura de contexto de retorno de chamada-campo BufferLength

Este é o comprimento em bytes do buffer de mensagem que foi transmitido para esta função.

O comprimento da mensagem real é fornecido no campo [DataLength](#) ..

O aplicativo pode usar o buffer inteiro para seus próprios propósitos durante a função de retorno de chamada.

Este é um campo de entrada para a função do consumidor de mensagens; não é relevante para uma função do manipulador de exceções.

#### CBCCALLBA (número inteiro assinado de 10 dígitos)

Estrutura de contexto de retorno de chamada-campo CallbackArea

Este é um campo que está disponível para a função de retorno de chamada usar.

O gerenciador de filas não toma nenhuma decisão com base no conteúdo desse campo e ele é transmitido inalterado a partir do campo CBD`CALLBA` na estrutura MQCBD, que é um parâmetro na chamada MQCB usado para definir a função de retorno de chamada..

Mudanças no *CB`CALLBA`* são preservadas nas chamadas da função de retorno de chamada para um *CB`CHOBJ`*. Este campo não é compartilhado com funções de retorno de chamada para outros identificadores

Este é um campo de entrada / saída para a função callback. O valor inicial desse campo é um ponteiro nulo ou bytes nulos

### **CB`CALLT` (número inteiro assinado de 10 dígitos)**

Estrutura de contexto de retorno de chamada-campo CallType .

Campo contendo informações sobre por que essa função foi chamada. Os tipos de chamada a seguir são definidos

Tipos de chamada de entrega de mensagem: esses tipos de chamada contêm informações sobre uma mensagem... Os parâmetros **CB`CLEN`** e **CB`BUFFLEN`** são válidos para esses tipos de chamada

#### **CB`TMR`**

A função de consumidor de mensagens foi chamada com uma mensagem que foi removida destrutivamente da manipulação de objetos.

Se o valor de *CB`CC`* for CCWARN, o valor do campo *Reason* será RC2079 ou um dos códigos indicando um problema de conversão de dados.

#### **CB`TMN`**

A função de consumidor de mensagens foi chamada com uma mensagem que ainda não foi removida destrutivamente do identificador de objetos. A mensagem pode ser removida destrutivamente da manipulação de objetos usando o *MsgToken*.

A mensagem pode não ter sido removida porque:

- As opções MQGMO solicitaram uma operação de navegação, GMBR\*
- A mensagem é maior que o buffer disponível e as opções MQGMO não especificam *gmattm*

Se o valor de *CB`CC`* for CCWARN, o valor do campo *Reason* será RC2080 ou um dos códigos indicando um problema de conversão de dados.

Tipos de chamada de controle de retorno de chamada: Esses tipos de chamada contêm informações sobre o controle do retorno de chamada e não contêm detalhes sobre uma mensagem.. Esses tipos de chamada são solicitados usando CBD`OPT` na estrutura MQCBD

Os parâmetros **CB`CLEN`** e **CB`BUFFLEN`** não são válidos para esses tipos de chamadas

#### **CB`TRC`**

O propósito deste tipo de chamada é permitir que a função de retorno de chamada execute alguma configuração inicial..

A função de retorno de chamada é chamada imediatamente após o retorno de chamada ser registrado, ou seja, após o retorno de uma chamada MQCB usando um valor para o campo *Operation* de CBREG.

Esse tipo de chamada é usado para consumidores de mensagens e manipuladores de eventos.

Se solicitado, esta é a primeira chamada da função de retorno de chamada..

O valor do campo *CB`CREA`* é RCNONE..

#### **CB`TSC`**

O propósito desse tipo de chamada é permitir que a função de retorno de chamada execute alguma configuração quando ela for iniciada, por exemplo, restabelecendo os recursos que foram limpos quando foram interrompidos anteriormente

A função de retorno de chamada é chamada quando a conexão é iniciada usando CTLSR ou CTL`SW`.

Se uma função de retorno de chamada for registrada dentro de outra função de retorno de chamada, esse tipo de chamada será chamado quando o retorno de chamada retornar.

Esse tipo de chamada é usado apenas para consumidores da mensagem

O valor do campo *CBCREA* é RCNONE..

#### **CBCTTC**

O propósito desse tipo de chamada é permitir que a função de retorno de chamada execute alguma limpeza quando ela for interrompida por um tempo, por exemplo, limpando recursos adicionais que foram obtidos durante o consumo de mensagens

A função de retorno de chamada é chamada quando uma chamada MQCTL é emitida usando um valor para o campo *Operation* de CTLSP.

Esse tipo de chamada é usado apenas para consumidores da mensagem

O valor do campo *CBCREA* é configurado para indicar o motivo da parada.

#### **CBCTDC**

O propósito deste tipo de chamada é permitir que a função de retorno de chamada execute a limpeza final no final do processo de consumo... A função de retorno de chamada é chamada quando:

- A função de retorno de chamada é cancelada usando uma chamada MQCB com BCUNR.
- A fila é fechada, causando um cancelamento implícito. Neste caso, a função de retorno de chamada é transmitida HOUNUH como a manipulação de objetos..
- Chamada MQDISC concluída-causando um fechamento implícito e, portanto, um cancelamento de registro. Nesse caso, a conexão não será desconectada imediatamente e qualquer transação em andamento ainda não será confirmada

Se qualquer uma dessas ações for executada dentro da própria função de retorno de chamada, a ação será chamada quando o retorno de chamada for retornado.

Esse tipo de chamada é usado para consumidores de mensagens e manipuladores de eventos.

Se solicitado, esta é a última chamada da função de retorno de chamada.

O valor do campo *CBCREA* é configurado para indicar o motivo da parada.

#### **CBCTEC**

##### **Função do manipulador de eventos**

A função do manipulador de eventos foi chamada sem uma mensagem quando:

- Uma chamada MQCTL é emitida com um valor para o campo *Operation* de CTLSP ou
- O gerenciador de filas ou a conexão para ou coloca em modo quiesce.

Essa chamada pode ser usada para executar a ação apropriada para todas as funções de retorno de chamada

##### **• Função do consumidor de mensagens**

A função do consumidor de mensagens foi chamada sem uma mensagem quando um erro (*CBCCC* = CCFAIL) foi detectado que é específico da manipulação de objetos; por exemplo, *CBCREA* code = RC2016 .

O valor do campo *CBCREA* é configurado para indicar o motivo da chamada.

Este é um campo de entrada. CBCTMR e CMCTMN são aplicáveis apenas às funções do consumidor de mensagem....

#### **CBCCC (número inteiro assinado de 10 dígitos)**

Estrutura de contexto de retorno de chamada-campo CompCode

Este é o código de conclusão. Indica se houve algum problema ao consumir a mensagem; ele é um dos seguintes:

**CCOK**

Conclusão bem-sucedida

**CCWARN**

Aviso (conclusão parcial)

**CCFAIL**

Chamada com falha

Este é um campo de entrada. O valor inicial desse campo é CCOK.

**CBCONNAREA (número inteiro assinado de 10 dígitos)**

Estrutura de contexto de retorno de chamada-campo ConnectionArea

Este é um campo que está disponível para a função de retorno de chamada usar.

O gerenciador de filas não toma decisões com base no conteúdo desse campo e é transmitido inalterado a partir do campo ConnectionArea na estrutura MQCTLO, que é um parâmetro na chamada MQCTL usado para controlar a função de retorno de chamada.

Quaisquer mudanças feitas nesse campo pelas funções de retorno de chamada são preservadas nas chamadas da função de retorno de chamadas. Essa área pode ser usada para transmitir informações que devem ser compartilhadas pelas funções de retorno de chamada. Ao contrário do *CallbackArea*, essa área é comum em todos os retornos de chamada para uma manipulação de conexões

Este é um campo de entrada e de saída O valor inicial desse campo é um ponteiro nulo ou bytes nulos

**CBCLLEN (número inteiro assinado de 10 dígitos)**

Este é o comprimento em bytes dos dados do aplicativo na mensagem. Se o valor for zero, significa que a mensagem não contém dados do aplicativo.

O campo CBCLLEN contém o comprimento da mensagem mas não necessariamente o comprimento dos dados da mensagem transmitidos ao consumidor. Pode ser que a mensagem tenha sido truncada. Use o campo GMRL no MQGMO para determinar quantos dados foram transmitidos para o consumidor.

Se o código de razão indicar que a mensagem foi truncada, você pode usar o campo CBCLLEN para determinar o tamanho da mensagem real.. Isso permite determinar o tamanho do buffer necessário para acomodar os dados da mensagem e, em seguida, emitir uma chamada MQCB para atualizar o CBDMML no MQCBD com um valor apropriado.

Se a opção GMCONV for especificada, a mensagem convertida poderá ser maior que o valor retornado para DataLength. Nesses casos, o aplicativo provavelmente precisa emitir uma chamada MQCB para atualizar o CBDMML no MQCBD para ser maior que o valor retornado pelo gerenciador de filas para DataLength.

Para evitar problemas de truncamento de mensagens, especifique o Comprimento MaxMsg como CBDFM Isso faz com que o gerenciador de filas aloque um buffer para o comprimento da mensagem integral após a conversão dos dados No entanto, esteja ciente de que, mesmo se essa opção for especificada, ainda é possível que armazenamento suficiente não esteja disponível para processar corretamente a solicitação.. Os aplicativos devem sempre verificar o código de razão retornado Por exemplo, se não for possível alocar armazenamento suficiente para converter a mensagem, as mensagens serão retornadas para o aplicativo não convertidas

Este é um campo de entrada para a função do consumidor de mensagens; não é relevante para uma função do manipulador de eventos.

**CBCFLG (número inteiro assinado de 10 dígitos)**

Sinalizadores que contêm informações sobre esse consumidor

A seguinte opção é definida:

### **CBCFBE**

Esse sinalizador pode ser retornado se uma chamada MQCLOSE anterior usando a opção COQSC falhar com um código de razão de RC2458

Este código indicou que a última mensagem de leitura antecipada está sendo retornada e que o buffer agora está vazio. Se o aplicativo emitir outra chamada MQCLOSE usando a opção COQSC, será bem-sucedido.

Observe que não é garantido que um aplicativo receberá uma mensagem com esse sinalizador configurado, pois ainda pode haver mensagens no buffer de leitura antecipada que não correspondem aos critérios de seleção atuais. Nessa instância, a função de consumidor é chamada com o código de razão RC2019 ..

Se o buffer de leitura antecipada estiver vazio, o consumidor será chamado com a sinalização CBCFBE e o código de razão RC2518.

Este é um campo de entrada para a função do consumidor de mensagens; não é relevante para uma função do manipulador de eventos.

### **CBCHOBJ (número inteiro assinado de 10 dígitos)**

Estrutura de contexto de retorno de chamada-campo CBCHOBJ

Para uma chamada para um consumidor de mensagens, este é o identificador para o objeto relacionado ao consumidor de mensagem

Para um manipulador de eventos, este valor é HONONE

O aplicativo pode usar esse identificador e o token de mensagem no bloco Obter Opções de Mensagem para obter a mensagem se uma mensagem não tiver sido removida da fila

Isto é sempre um campo de entrada. O valor inicial deste campo é HOUNUH

### **CBCRCD (número inteiro assinado de 10 dígitos)**

**CBCRCD** indica por quanto tempo o gerenciador de filas espera antes de tentar se reconectar. O campo pode ser modificado por um manipulador de eventos para alterar o atraso ou parar a reconexão.

Use o campo **CBCRCD** apenas se o valor do campo **Reason** no contexto de retorno de chamada for RC2545.

Na entrada para o manipulador de eventos, o valor de **CBCRCD** é o número de milissegundos que o gerenciador de fila aguardará antes de fazer uma tentativa de reconexão.. [Tabela 685 na página 1059](#) lista os valores que podem ser configurados para modificar o comportamento do gerenciador de filas no retorno do manipulador de eventos

<i>Tabela 685. CBCRCD valores</i>	
<b>Value</b>	<b>Descrição</b>
-1	Não faça mais tentativas de reconexão. Um erro é retornado para o aplicativo
0	Tente reconectar imediatamente.
>0	Aguarde por esses milissegundos antes de tentar a conexão novamente.

### **CBCREA (número inteiro assinado de 10 dígitos)**

Estrutura do contexto de retorno de chamada-campo Motivo

Este é o código de razão que qualifica o CBCCC

Este é um campo de entrada. O valor inicial deste campo é RCNONE..

### **CBCSTATE (número inteiro assinado de 10 dígitos)**

Uma indicação do estado do consumidor atual. Este campo é de maior valor para um aplicativo quando um código de razão diferente de zero é transmitido para a função do consumidor...

É possível usar esse campo para simplificar a programação de aplicativos, pois não é necessário codificar o comportamento para cada código de razão

Este é um campo de entrada. O valor inicial deste campo é CSNONE

<i>Tabela 686. Valores de CBCSTATE e ações resultantes</i>		
<b>Estado</b>	<b>Ação do gerenciador de filas</b>	<b>Valor da constante</b>
<i>CSNONE</i> Este código de razão representa uma chamada normal sem informações de razão adicionais	Nenhum; esta é a operação normal	0
<i>CSSUST</i> Esses códigos de razão representam condições temporárias..	A rotina de retorno de chamada é chamada para relatar a condição e, em seguida, suspensa Após um período, o sistema pode tentar a operação novamente, o que pode levar à mesma condição sendo levantada novamente.	1
<i>CSSUSU</i> Esses códigos de razão representam as condições nas quais o retorno de chamada precisa agir para resolver a condição	O consumidor é suspenso, e a rotina de retorno de chamada é chamada para relatar a condição A rotina de retorno de chamada deve resolver a condição se possível e RESUME ou fechar a conexão.	2
<i>CSSUS</i> Esses códigos de razão representam falhas que evitam retornos de chamada de mensagens adicionais	O gerenciador de filas suspende automaticamente a função de retorno de chamada Se a função de retorno de chamada for retomada, provavelmente receberá o mesmo código de razão novamente.	3
<i>CSSTOP</i> Esses códigos de razão representam o término do consumo de mensagens	Entregue para o manipulador de exceções e para retornos de chamadas que especificaram CBDTC Nenhuma mensagem adicional pode ser consumida	4

### **CBCSID (número inteiro assinado de 10 dígitos)**

Estrutura do contexto de retorno de chamada-campo StrucId

Este é o identificador de estrutura; o valor deve ser:

#### **CBCSI**

Identificador para estrutura de contexto de retorno de chamada.

Isto é sempre um campo de entrada. O valor inicial deste campo é o CBCSI.

### **CBCVER (número inteiro assinado de 10 dígitos).**

Estrutura de contexto de retorno de chamada-Campo de versão

Este é o número da versão da estrutura; o valor deve ser:

#### **CBCV1**

Estrutura de contexto de retorno de chamada Version-1 .

A constante a seguir especifica o número da versão atual:

#### **CBCCV**

A versão atual da estrutura de contexto de retorno de chamada



Isto é sempre um campo de entrada. O valor inicial desse campo é CBCV1.

## Valores iniciais

<i>Tabela 687. Campos em MQCBC</i>		
<b>Nome de Campo</b>	<b>Nome da constante</b>	<b>Valor da constante</b>
<i>CBCSID</i>	CBCSI	'CBC↵'
<i>CBCVER</i>	CBCV1	1
<i>CBCCALLT</i>	Nenhum	0
<i>CBCHOBJ</i>	HOUNUH	-1
<i>CBCCALLBA</i>	Nenhum	Ponteiro nulo ou bytes nulos
<i>CBCCONNAREA</i>	Nenhum	Ponteiro nulo ou bytes nulos
<i>CBCCC</i>	CCOK	0
<i>CBCREA</i>	RCNONE	0
<i>CBCSTATE</i>	CSNONE	0
<i>CBCLLEN</i>	Nenhum	0
<i>CBCBUFFLEN</i>	Nenhum	0
<i>CBCFLG</i>	Nenhum	0
<i>CBCRCD</i>	Nenhum	0

### Nota:

1. O símbolo ↵ representa um único caractere em branco.

## Declaração RPG

```

D* MQCBC Structure
D*
D*
D* Structure identifier
D  CBCSID          1      4    INZ('CBC ')
D*
D* Structure version number
D  CBCVER          5      8I 0 INZ(1)
D*
D* Why Function was called
D  CBCCALLT       9      12I 0 INZ(0)
D*
D* Object Handle
D  CBCHOBJ       13     16I 0 INZ(-1)
D*
D* Callback data passed to the function
D  CBCCALLBA     17     32*  INZ(*NULL)
D*
D* MQCTL Data area passed to the function
D  CBCCONNAREA   33     48*  INZ(*NULL)
D*
D* Completion Code
D  CBCCC         49     52I 0 INZ(0)
D*
D* Reason Code
D  CBCREA        53     56I 0 INZ(0)
D*
D* Consumer State
D  CBCSTATE      57     60I 0 INZ(0)

```

```

D*
D* Message Data Length
D  CBCLEN          61      64I 0 INZ(0)
D*
D* Buffer Length
D  CBCBUFFLEN     65      68I 0 INZ(0)
D*
** Flags containing information about
D* this consumer
D  CBCFLG          69      72I 0 INZ(0)
D* Ver:1 **
D* Number of milliseconds before reconnect attempt
D  CBCRCD          73      76I 0 INZ(0)
D* Ver:2 **
D*

```

## IBM i MQCBD (Descritor de retorno de chamada) em IBM i

Estrutura especificando a função de retorno de chamada.

### Visão Geral

**Propósito:** A estrutura MQCBD é usada para especificar uma função de retorno de chamada e as opções que controlam seu uso pelo gerenciador de filas.

A estrutura é um parâmetro de entrada na chamada MQCB

**Versão:** a versão atual do MQCBD é CBDV1

**Conjunto de caracteres e codificação:** os dados no MQCBD devem estar no conjunto de caracteres e na codificação do gerenciador de filas locais; eles são fornecidos pelo atributo do gerenciador de filas **CodedCharSetId** e ENNAT. No entanto, se o aplicativo estiver em execução como um IBM MQ MQI client, a estrutura deverá estar no conjunto de caracteres e na codificação do cliente.

- [“Campos” na página 1062](#)
- [“Valores iniciais” na página 1066](#)
- [“Declaração RPG” na página 1066](#)

### Campos

A estrutura MQCBD contém os campos a seguir; os campos são descritos em **ordem alfabética**:

#### **CBDCALLBA (número inteiro assinado de 10 dígitos)**

Este é um campo que está disponível para a função de retorno de chamada usar.

O gerenciador de filas não toma decisões com base no conteúdo desse campo e ele é transmitido inalterado do campo [CBCCALLBA](#) na estrutura MQCBD, que é um parâmetro na declaração da função de retorno de chamada.

O valor é usado apenas em um *Operation* que tenha um valor CBREG, sem retorno de chamada definido atualmente, ele não substitui uma definição anterior...

Este é um campo de entrada e saída para a função de retorno de chamada. O valor inicial desse campo é um ponteiro nulo ou bytes nulos

#### **CBDCALLBF (número inteiro assinado de 10 dígitos)**

A função de retorno de chamadas é chamada como uma chamada de função.

Use este campo para especificar um ponteiro para a função de retorno de chamada..

Você deve especificar *CallbackFunction* ou *CallbackName*. Se você especificar ambos, o código de razão RC2486 será retornado..

Se nem *CallbackName* nem *CallbackFunction* não for configurado, a chamada falhará com o código de razão RC2486

Essa opção não é suportada nos seguintes ambientes:

- CICS no z/OS
  - Linguagens de programação e compiladores que não suportam referências de ponteiro de funções
- Nessas situações, a chamada falha com o código de razão RC2486.

Este é um campo de entrada. O valor inicial desse campo é um ponteiro nulo ou bytes nulos

### **CBDCALLBN (número inteiro assinado de 10 dígitos)**

A função callback é chamada como um programa dinamicamente vinculado.

Você deve especificar *CallbackFunction* ou *CallbackName*. Se você especificar ambos, o código de razão RC2486 será retornado..

Se *CallbackName* ou *CallbackFunction* não for true, a chamada falhará com o código de razão RC2486..

O módulo é carregado quando a primeira rotina de retorno de chamada a ser usada é registrada e descarregado quando a última rotina de retorno de chamada para usá-la remove o registro.

Exceto onde indicado no texto a seguir, o nome é alinhado à esquerda dentro do campo, sem espaços em branco integrados; o próprio nome é preenchido com espaços em branco para o comprimento do campo. Nas descrições a seguir, os colchetes ([]) denotam informações opcionais:

#### **IBMi**

O nome do retorno de chamadas pode ser um dos seguintes formatos:

- Programa de Biblioteca "/"
- Biblioteca "/" ServiceProgram ("FunctionName")

Por exemplo, MyLibrary/MyProgram(MyFunction).

O nome da biblioteca pode ser \*LIBL Ambos os nomes de biblioteca e de programa são limitados a um máximo de 10 caracteres.

#### **AIX and Linux**

O nome do retorno de chamada é o nome de um módulo ou biblioteca carregável dinamicamente, sufixado com o nome de uma função residente nessa biblioteca. O nome da função deve estar entre parênteses. Opcionalmente, o nome da biblioteca pode ser prefixado com um caminho do diretório:

```
[path]library(function)
```

Se o caminho não for especificado, o caminho da procura do sistema será usado

O nome é limitado a no máximo 128 caracteres.

#### **Windows**

O nome do retorno de chamada é o nome de uma biblioteca de link dinâmico, sufixado com o nome de uma função residente nessa biblioteca. O nome da função deve estar entre parênteses. O nome da biblioteca pode, opcionalmente, ser prefixado com um caminho de diretório e unidade:

```
[d:][path]library(function)
```

Se a unidade e o caminho não forem especificados, o caminho da procura do sistema será usado

O nome é limitado a no máximo 128 caracteres.

#### **z/OS**

O nome do retorno de chamada é o nome de um módulo de carregamento que é válido para a especificação no parâmetro EP da macro LINK ou LOAD

O nome é limitado a um máximo de 8 caracteres.

## **z/OS CICS**

O nome do retorno de chamada é o nome de um módulo de carregamento válido para especificação no parâmetro PROGRAM da macro do comando EXEC CICS LINK.

O nome é limitado a um máximo de 8 caracteres.

O programa pode ser definido como remoto utilizando a opção REMOTESYTEM da definição PROGRAM instalada ou pelo programa de roteamento dinâmico.

A região CICS remota deve ser conectada ao IBM MQ se o programa for usar chamadas API do IBM MQ . No entanto, observe que o campo CBCHOBJ na estrutura MQCBC não é válido em um sistema remoto.

Se ocorrer uma falha ao tentar carregar *CaLLbackName*, um dos códigos de erro a seguir será retornado para o aplicativo:

- RC2495
- RC2496
- RC2497

Uma mensagem também é gravada no log de erros que contém o nome do módulo para o qual o carregamento foi tentado e o código de razão com falha do sistema operacional

Este é um campo de entrada. O valor inicial deste campo é uma cadeia nula ou espaços em branco.

## **CBDCALLBT (número inteiro assinado de 10 dígitos)**

Este é o tipo da função de retorno de chamada.. O valor deve ser um de:

### **CBTMC**

Define esse retorno de chamada como uma função do consumidor de mensagens..

Uma função de retorno de chamada do consumidor de mensagens é chamada quando uma mensagem, atendendo aos critérios de seleção especificados, está disponível em um identificador de objeto e a conexão é iniciada.

### **CBTEH**

Define esse retorno de chamada como a rotina de evento assíncrono; ele não é orientado a consumir mensagens para um identificador.

O *Hobj* não é necessário na chamada MQCB que define o manipulador de eventos e será ignorado se especificado

O manipulador de eventos é chamado para condições que afetam todo o ambiente do consumidor de mensagens A função do consumidor é chamada sem uma mensagem quando ocorre um evento, por exemplo, um gerenciador de filas ou parada de conexão ou quiesce. Ele não é chamado para condições que são específicas para um único consumidor de mensagens, por exemplo, RC2016

Os eventos são entregues para o aplicativo, independentemente de a conexão ser iniciada ou interrompida, exceto nos ambientes a seguir:

- CICS no ambiente z/OS
- aplicativos não encadeados

Se o responsável pela chamada não passar um desses valores, a chamada falhará com um código de razão de RC2483 .

Isto é sempre um campo de entrada. O valor inicial deste campo é CBTMC..

## **CBDMML (número inteiro assinado de 10 dígitos)**

Este é o comprimento em bytes da mensagem mais longa que pode ser lida a partir do identificador e fornecida à rotina de retorno de chamada. Se uma mensagem tiver um comprimento maior, a rotina de retorno de chamada receberá *MaxMsgLength* bytes da mensagem e código de razão:

- RC2080 ou

- RC2079 se tiver especificado GMATM.

O comprimento real da mensagem é fornecido no campo “CBCLLEN (número inteiro assinado de 10 dígitos)” na página 1058 da estrutura do MQCBC

O seguinte valor especial é definido:

#### **CBDFM**

O comprimento do buffer é ajustado pelo sistema para retornar as mensagens sem truncamento.

Se memória insuficiente estiver disponível para alocar um buffer para receber a mensagem, o sistema chama a função de retorno de chamada com um código de razão RC2071 .

Se, por exemplo, você solicitar a conversão de dados e houver memória insuficiente disponível para converter os dados da mensagem, a mensagem não convertida será transmitida para a função de retorno de chamada.

Este é um campo de entrada. O valor inicial do campo *MaxMsgLength* é CBDFM.

#### **CBDOPT (número inteiro assinado de 10 dígitos)**

Estrutura do descritor de retorno de chamada-campo Opções.

Qualquer um ou todos os itens a seguir podem ser especificados. Para especificar mais de uma opção, inclua os valores juntos (não inclua a mesma constante mais de uma vez) ou combine os valores usando a operação OR bit a bit (se a linguagem de programação suportar operações de bit). As combinações que não são válidas são observadas; quaisquer outras combinações são válidas

#### **CBDFQ**

A chamada MQCB falhará se o gerenciador de filas estiver no estado de quiesce

No z/OS, essa opção também força a chamada MQCB a falhar se a conexão (para um aplicativo CICS ou IMS ) estiver no estado de quiesce

Especifique GMFIQ, nas opções MQGMO transmitidas na chamada MQCB, para causar notificação aos consumidores de mensagens quando eles estiverem quiesce.

**Opções de controle:** As opções a seguir controlam se a função de retorno de chamada é chamada, sem uma mensagem, quando o estado do consumidor é alterado:

#### **CBDRC**

A função de retorno de chamada é chamada com o tipo de chamada CBCTRC

#### **CBDSC**

A função callback é chamada com o tipo de chamada CBCTSC.

#### **CBDTC**

A função de retorno de chamada é chamada com o tipo de chamada CBCTTC

#### **CBDDC**

A função de retorno de chamada é chamada com o tipo de chamada CBCTDC

Consulte “CBCCALLT (número inteiro assinado de 10 dígitos)” na página 1056 para obter detalhes adicionais sobre esses tipos de chamada..

**Opção padrão:** se você não precisar de nenhuma das opções descritas, use a seguinte opção:

#### **CBDNO**

Use esse valor para indicar que nenhuma outra opção foi especificada. Todas as opções assumem seus valores-padrão.

CBDNO é definido para auxiliar a documentação do programa; não se pretende que esta opção seja usada com qualquer outra, mas como seu valor é zero, tal uso não pode ser detectado.

Este é um campo de entrada. O valor inicial do campo *Options* é CBDNO.

#### **CBDSID (número inteiro assinado de 10 dígitos)**

Estrutura do descritor de retorno de chamada-campo StrucId .

Este é o identificador de estrutura; o valor deve ser:

#### **CBDSI**

Identificador para a estrutura do descritor de retorno de chamada

Isto é sempre um campo de entrada. O valor inicial desse campo é CBDSI..

#### **CBDVER (número inteiro assinado de 10 dígitos)**

Estrutura do descritor de retorno de chamada-campo Versão.

Este é o número da versão da estrutura; o valor deve ser:

#### **CBDV1**

Estrutura do descritor de retorno de chamada Version-1

A constante a seguir especifica o número da versão atual:

#### **CBDCV**

Versão atual da estrutura do descritor de callback.

Isto é sempre um campo de entrada. O valor inicial desse campo é CBDV1.

### **Valores iniciais**

<b>Nome de Campo</b>	<b>Nome da constante</b>	<b>Valor da constante</b>
<i>StrucId</i>	CBDSI	'CBD-'
<i>Version</i>	CBDV1	1
<i>CallBackType</i>	CBTMC	1
<i>Options</i>	CBDNO	0
<i>CallBackArea</i>	Nenhum	Bytes nulos
<i>CallBackFunction</i>	Nenhum	Bytes nulos
<i>CallBackName</i>	Nenhum	Espaços em branco
<i>MaxMsgLength</i>	CBD FM	-1

#### **Nota:**

1. O símbolo - representa um único caractere em branco.

### **Declaração RPG**

```
D* MQCBD Structure
D*
D*
D* Structure identifier
D  CBDSID          1      4    INZ('CBD ')
D*
D* Structure version number
D  CBDVER          5      8I 0 INZ(1)
D*
D* Callback function type
D  CBDCALLBT      9      12I 0 INZ(1)
D*
** Options controlling message
D* consumption
D  CBDOPT         13     16I 0 INZ(0)
D*
D* User data passed to the function
D  CBDCALLBA     17     32*
D*
D* FP: Callback function pointer
```

D	CBDCALLBF	33	48*	
D*	Callback name			
D	CBDCALLBN	49	176	INZ('\0')
D*	Maximum message length			
D	CBDMML	177	180I 0	INZ(-1)



## MQCHARV (Sequência de Comprimento Variável) no IBM i

Use a estrutura MQCHARV para descrever uma sequência de comprimento variável.

### Visão Geral

**Conjunto de caracteres e codificação:** os dados no MQCHARV devem estar na codificação do gerenciador de filas locais que é fornecido pelo ENNAT e o conjunto de caracteres do campo VCHRC na estrutura. Se o aplicativo estiver em execução como um IBM MQ MQI client, a estrutura deverá estar na codificação do cliente. Alguns conjuntos de caracteres possuem uma representação que depende da codificação... Se VCHRC for um desses conjuntos de caracteres, a codificação usada será a mesma codificação dos outros campos no MQCHARV. O conjunto de caracteres identificado por VSCCSID pode ser um conjunto de caracteres de duplo byte (DBCS).

**Uso:** a estrutura MQCHARV endereça dados que podem ser descontíguos com a estrutura que os contém. Para tratar esses dados, campos declarados com o tipo de dados do ponteiro podem ser utilizados.

- [“Campos” na página 1067](#)
- [“Valores iniciais” na página 1068](#)
- [“Declaração RPG” na página 1069](#)
- [“Redefinição de CSAPL” na página 1069](#)

### Campos

A estrutura MQCHARV contém os seguintes campos; os campos são descritos em **ordem alfabética**:

#### VCHRC (número inteiro assinado de 10 dígitos)

Esse é o identificador do conjunto de caracteres da cadeia de comprimento variável endereçada pelo campo VCHRP ou VCHRO..

O valor inicial desse campo é CSAPL. Isso é definido por IBM MQ para indicar que ele deve ser alterado pelo gerenciador de filas para o identificador do conjunto de caracteres verdadeiro do gerenciador de filas.. Isso é da mesma maneira que o CSQM se comporta. Como resultado, o valor CSAPL nunca é associado a uma sequência de comprimento variável.. O valor inicial desse campo pode ser alterado definindo um valor diferente para a constante CSAPL para sua unidade de compilação pelos meios apropriados para a linguagem de programação de seu aplicativo.

#### VCHRL (número inteiro assinado de 10 dígitos)

O comprimento em bytes da cadeia de comprimento variável endereçada pelo campo VCHRP ou VCHRO.

O valor inicial desse campo é 0. O valor deve ser maior ou igual a zero ou o valor especial a seguir que é reconhecido:

#### VSFLT

Se VSFLT não for especificado, os bytes VCHRL serão incluídos como parte da sequência. Se caracteres nulos estiverem presentes, eles não delimitarão a sequência.

Se VSFLT for especificado, a cadeia será delimitada pelo primeiro nulo encontrado na cadeia. O próprio nulo não é incluído como parte dessa cadeia.

**Nota:** O caractere nulo utilizado para terminar uma cadeia se o VSFLT for especificado é nulo do conjunto de códigos especificado pelo VCHRC.

Por exemplo, em UTF-16 (CCSIDs 1200, 13488 e 17584), esta é a codificação Unicode de 2 bytes em que um nulo é representado por um número de 16 bits de todos os zeros. Em UTF-16, é comum localizar bytes únicos configurados para todos os zero que fazem parte de caracteres (caracteres ASCII de 7 bits, por exemplo), mas as sequências serão terminadas em nulo apenas quando dois bytes 'zero' forem localizados em um limite de bytes pares. É possível obter dois bytes 'zero' em um limite ímpar quando cada parte de caracteres válidos. Por exemplo, x'01' x'00' x'00' x'30' representa dois caracteres Unicode válidos e não finaliza a sequência de caracteres.

### **VCHRO (número inteiro assinado de 10 dígitos)**

O deslocamento em bytes da sequência de comprimento variável a partir do início do MQCHARV ou da estrutura que o contém.

Quando a estrutura MQCHARV é integrada em outra estrutura, esse valor é o deslocamento em bytes da sequência de comprimento variável do início da estrutura que contém essa estrutura MQCHARV. Quando a estrutura MQCHARV não está integrada dentro de outra estrutura, por exemplo, se ela for especificada como um parâmetro em uma chamada de função, o deslocamento será relativo ao início da estrutura MQCHARV

O deslocamento pode ser positivo ou negativo. Você pode utilizar o campo VCHRP ou VCHRO para especificar a cadeia de comprimento variável, mas não ambos..

O valor inicial desse campo é 0.

### **VCHRP (ponteiro)**

Este é um ponteiro para a cadeia de comprimento variável..

Você pode utilizar o campo VCHRP ou VCHRO para especificar a cadeia de comprimento variável, mas não ambos..

O valor inicial desse campo é um ponteiro nulo ou bytes nulos

### **VCHRS (número inteiro assinado de 10 dígitos)**

O tamanho em bytes do buffer endereçado pelo campo VCHRP ou VCHRO.

Quando a estrutura MQCHARV é usada como um campo de saída em uma chamada de função, esse campo deve ser inicializado com o comprimento do buffer fornecido... Se o valor de VCHRL for maior que VCHRS, então apenas os bytes de dados VCHRS serão retornados ao responsável pela chamada no buffer.

O valor deve ser maior ou igual a zero ou o seguinte valor especial que é reconhecido:

#### **VSUSL**

Se VSUSL for especificado, o comprimento do buffer será obtido do campo VCHRL na estrutura MQCHARV. Esse valor especial não é apropriado quando a estrutura é usada como um campo de saída e um buffer é fornecido.. Esse é o valor inicial desse campo.

## **Valores iniciais**

<i>Tabela 689. Valores iniciais MQCHARV para constantes</i>		
<b>Nome de Campo</b>	<b>Nome da constante</b>	<b>Valor da constante</b>
VCHRP	Nenhum	Ponteiro nulo ou bytes nulos.
VCHRO	Nenhum	0
VCHRS	VSUSL	-1
VCHRL	Nenhum	0
VCHRC	CSAPL	-3



## Declaração RPG

```
D*..1....:....2....:....3....:....4....:....5....:....6....:....7..
D* MQCHARV Structure
D*
D* Address of variable length string
D VCHRP          1      16*
D* Offset of variable length string
D VCHRO          17     20I 0
D* Size of buffer
D VCHRS          21     24I 0
D* Length of variable length string
D VCHRL          25     28I 0
D* CCSID of variable length string
D VCHRC          29     32I 0
```

## Redefinição de CSAPL

Ao contrário das linguagens de programação suportadas em outras plataformas, o RPG não tem uma maneira de redefinir uma constante definida, portanto, você deve configurar cada VCHRC especificamente se desejar usar um valor diferente de CSAPL.

## MQCIH (cabeçalho CICS bridge) no IBM i

A estrutura MQCIH descreve as informações que podem estar presentes no início de uma mensagem enviada para o CICS bridge por IBM MQ for z/OS.

## Visão Geral

**Nome do formato:** FMCICS.

**Versão:** a versão atual do MQCIH é CIVER2. Campos que existem apenas na versão mais recente da estrutura são identificados como tal nas descrições a seguir.

O arquivo COPY fornecido contém a versão mais recente de MQCIH, com o valor inicial do campo *CIVER* configurado como CIVER2..

**Conjunto de caracteres e codificação:** condições especiais se aplicam ao conjunto de caracteres e à codificação usados para a estrutura MQCIH e dados da mensagem do aplicativo:

- Os aplicativos que se conectarem ao gerenciador de filas que possui a fila do CICS bridge devem fornecer uma estrutura MQCIH que esteja no conjunto de caracteres e na codificação do gerenciador de fila. Isso ocorre porque a conversão de dados da estrutura MQCIH não é executada neste caso.
- Os aplicativos que se conectam a outros gerenciadores de filas podem fornecer uma estrutura MQCIH que esteja em qualquer um dos conjuntos de caracteres e codificações suportados; a conversão do MQCIH é executada pelo agente do canal de mensagem de recebimento conectado ao gerenciador de filas que possui a fila do CICS bridge ..

**Nota:** Há uma exceção para isso. Se o gerenciador de filas que possui a fila CICS bridge estiver usando CICS para enfileiramento distribuído, o MQCIH deverá estar no conjunto de caracteres e na codificação do gerenciador de filas que possui a fila CICS bridge .

- Os dados da mensagem do aplicativo após a estrutura MQCIH devem estar no mesmo conjunto de caracteres e na codificação que a estrutura MQCIH. Os campos *CICSI* e *CIENC* na estrutura MQCIH não podem ser usados para especificar o conjunto de caracteres e a codificação dos dados da mensagem do aplicativo.

Uma saída de conversão de dados deve ser fornecida pelo usuário para converter os dados de mensagem do aplicativo se os dados não forem um dos formatos integrados suportados pelo gerenciador de filas.

**Uso:** se os valores necessários pelo aplicativo forem os mesmos que os valores iniciais mostrados em [Tabela 691 na página 1079e](#) a ponte estiver em execução com AUTH=LOCAL ou AUTH=IDENTIFY, a

estrutura MQCIH pode ser omitida da mensagem. Em todos os outros casos, a estrutura deve estar presente

A ponte aceita uma estrutura MQCIH version-1 ou version-2 , mas para transações 3270 uma estrutura version-2 deve ser usada.

O aplicativo deve assegurar que os campos documentados como campos de "solicitação" tenham valores apropriados na mensagem enviada para a ponte; esses campos são inseridos na ponte.

Os campos documentados como "resposta" são configurados pelo CICS bridge na mensagem de resposta que a ponte envia para o aplicativo. As informações de erro são retornados nos campos *CIRET*, *CIFNC*, *CICC*, *CIREA* e *CIAC* , mas nem todos eles são configurados em todos os casos A [Tabela 690 na página 1070](#) mostra quais campos são configurados para diferentes valores de *CIRET*

<i>Tabela 690. Conteúdo de campos de informações de erro na estrutura MQCIH..</i>				
<b>CIRET</b>	<b>CIFNC</b>	<b>CICC</b>	<b>CIREA</b>	<b>CIAC</b>
CRC000	-	-	-	-
CRC003	-	-	FBC*	-
CRC002 CRC008	Nome da chamada IBM MQ	IBM MQ <i>CMPCOD</i>	IBM MQ <i>REASON</i>	-
CRC001 CRC006 CRC007 CRC009	CICS EIBFN	CICS EIBRESP	CICS EIBRESP2	-
CRC004 CRC005	-	-	-	CICS ABCODE

- “Campos” na [página 1070](#)
- “Valores iniciais” na [página 1079](#)
- “Declaração RPG” na [página 1080](#)

## Campos

A estrutura MQCIH contém os seguintes campos; os campos são descritos em **ordem alfabética**:

### **CIAC (sequência de caracteres de 4 bytes)**

Código de finalização anormal de tarefa.

O valor retornado nesse campo é significativo apenas se o campo *CIRET* tiver o valor CRC005 ou CRC004. Se houver, *CIAC* conterá o valor CICS ABCODE.

Este é um campo de resposta. O comprimento deste campo é fornecido pelo LNABNC O valor inicial deste campo é 4 caracteres em branco.

Este é um indicador que especifica se os descritores do ADS devem ser enviados em solicitações SEND e RECEIVE BMS Os seguintes valores são definidos:

#### **ADNONE**

Não enviar ou receber descritor ADS.

#### **ADSEND**

Envie o descritor do ADS

#### **ADRECV**

Receber descritor ADS.

#### **ADMSGF**

Utilize o formato da mensagem para o descritor do ADS

Isso faz com que o descritor ADS seja enviado ou recebido usando o formato longo do descritor ADS. O formulário longo possui campos alinhados em limites de 4 bytes.

O campo *CIADS* deve ser configurado conforme a seguir:

- Se os descritores do ADS não estiverem sendo usados, configure o campo como ADNONE
- Se os descritores ADS *estiverem sendo usados* e com o *mesmo* CCSID em cada ambiente, configure o campo para a soma de ADSEND e ADRECV.
- Se os descritores ADS *estiverem sendo usados* , mas com CCSIDs *diferentes* em cada ambiente, configure o campo para a soma de ADSEND, ADRECV e ADMSGF.

Este é um campo de pedido utilizado apenas para transações 3270 O valor inicial deste campo é ADNONE

### **CIADS (número inteiro assinado de 10 dígitos)**

Enviar / receber descritor ADS.

Este é um indicador que especifica se os descritores do ADS devem ser enviados em solicitações SEND e RECEIVE BMS Os seguintes valores são definidos:

#### **ADNONE**

Não enviar ou receber descritor ADS.

#### **ADSEND**

Envie o descritor do ADS

#### **ADRECV**

Receber descritor ADS.

#### **ADMSGF**

Utilize o formato da mensagem para o descritor do ADS

Isso faz com que o descritor ADS seja enviado ou recebido usando o formato longo do descritor ADS. O formulário longo possui campos alinhados em limites de 4 bytes.

O campo *CIADS* deve ser configurado conforme a seguir:

- Se os descritores do ADS não estiverem sendo usados, configure o campo como ADNONE
- Se os descritores ADS *estiverem sendo usados* e com o *mesmo* CCSID em cada ambiente, configure o campo para a soma de ADSEND e ADRECV.
- Se os descritores ADS *estiverem sendo usados* , mas com CCSIDs *diferentes* em cada ambiente, configure o campo para a soma de ADSEND, ADRECV e ADMSGF.

Este é um campo de pedido utilizado apenas para transações 3270 O valor inicial deste campo é ADNONE

### **CIAI (sequência de caracteres de 4 bytes).**

Chave do AID.

Este é o valor inicial da chave AID quando a transação é iniciada. É um valor de um byte, alinhado à esquerda.

Este é um campo de pedido utilizado apenas para transações 3270 O comprimento desse campo é fornecido por LNATID. O valor inicial deste campo é 4 espaços em branco.

### **CIAUT (sequência de caracteres de 8 bytes).**

Senha ou passagem.

É uma senha ou passticket. Se a autenticação do identificador de usuário estiver ativa para o CICS bridge, *CIAUT* será usado com o identificador de usuário no contexto de identidade do MQMD para autenticar o emissor da mensagem.

Este é um campo de solicitação O comprimento deste campo é fornecido por LNAUTH O valor inicial deste campo é 8 espaços em branco.

### **CICC (número inteiro assinado de 10 dígitos).**

código de conclusão IBM MQ ou CICS EIBRESP.

O valor retornado nesse campo é dependente de *CIRET* ; Consulte [Tabela 690 na página 1070](#).

Este é um campo de resposta. O valor inicial desse campo é CCOK.

#### **CICNC (sequência de caracteres de 4 bytes)**

Código de transação de encerramento de forma anormal

Este é o código de encerramento anormal a ser usado para finalizar a transação (normalmente uma transação conversacional que está solicitando mais dados). Caso contrário, esse campo será configurado como espaços em branco

Este é um campo de pedido utilizado apenas para transações 3270 O comprimento deste campo é fornecido por LNCNCL O valor inicial deste campo é 4 espaços em branco.

#### **CICP (inteiro assinado de 10 dígitos)**

Posição do cursor.

Esta é a posição inicial do cursor quando a transação é iniciada Posteriormente, para transações conversacionais, a posição do cursor está no vetor RECEIVE.

Este é um campo de pedido utilizado apenas para transações 3270 O valor inicial desse campo é 0. Esse campo não estará presente se *CIVER* for menor que *CIVER2*.

#### **CICSI (número inteiro assinado de 10 dígitos)**

Reservado.

Este é um campo reservado; seu valor não é significativo. O valor inicial desse campo é 0.

#### **CICT (número inteiro assinado de 10 dígitos)**

Se a tarefa pode ser de conversação

Este é um indicador que especifica se a tarefa deve ter permissão para emitir solicitações para obter mais informações ou se deve ser encerrada de forma anormal. O valor deve ser um dos seguintes:

##### **CTYES**

Tarefa é conversacional.

##### **CTNO**

Tarefa não é conversacional.

Este é um campo de pedido utilizado apenas para transações 3270 O valor inicial deste campo é CTNO..

#### **CIENC (número inteiro assinado de 10 dígitos)**

Reservado.

Este é um campo reservado; seu valor não é significativo. O valor inicial desse campo é 0.

#### **CIEO (número inteiro assinado de 10 dígitos).**

Compensação de erro na mensagem.

Esta é a posição de dados inválidos detectada pela saída da ponte Este campo fornece o deslocamento do início da mensagem para o local dos dados inválidos.

Este é um campo de resposta utilizado apenas para transações 3270 O valor inicial desse campo é 0. Esse campo não estará presente se *CIVER* for menor que *CIVER2*.

#### **CIFAC (cadeia de bits de 8 bytes)**

Token do recurso de ponte

Este é um token de recurso de ponte de 8 bytes O propósito de um token de recurso de ponte é permitir que várias transações em uma pseudoconversação usem o mesmo recurso de ponte (terminal 3270 virtual).. Na primeira, ou única, mensagem em uma pseudoconversação, um valor de FCNONE deve ser configurado; isso informa ao CICS para alocar um novo recurso de ponte para essa mensagem... Um token de recurso de Ponte é retornado em mensagens de resposta quando

um *CIFKT* diferente de zero é especificado na mensagem de entrada. As mensagens de entrada subsequentes podem então usar o mesmo token do recurso de ponte.

O seguinte valor especial é definido:

**FCNONE**

Nenhum token BVT especificado.

Este é um campo de solicitação e de resposta utilizado apenas para transações 3270. O comprimento deste campo é fornecido por LNFAC O valor inicial deste campo é FCNONE.

**CIFKT (número inteiro assinado de 10 dígitos)**

Tempo de liberação da instalação de ponte

Este é o período de tempo em segundos que o recurso de ponte será mantido após a transação do usuário ter terminado. Para transações não conversacionais, o valor deve ser zero.

Este é um campo de pedido utilizado apenas para transações 3270 O valor inicial desse campo é 0.

**CIFL (sequência de caracteres de 4 bytes)**

Atributos emulados do Terminal

Este é o nome de um terminal instalado que deve ser usado como um modelo para o recurso de ponte Um valor de espaços em branco significa que *CIFL* é obtido da definição de perfil de transação de ponte ou um valor padrão é usado...

Este é um campo de pedido utilizado apenas para transações 3270 O comprimento deste campo é fornecido por LNFACL O valor inicial deste campo é 4 espaços em branco.

**CIFLG (número inteiro assinado de 10 dígitos)**

Sinalizadores.

O valor deve ser:.

**CIFNON**

Sem bandeiras.

Este é um campo de solicitação O valor inicial desse campo é CIFNON.

**CIFMT (cadeia de caracteres de 8 bytes)**

O nome do formato IBM MQ de dados que seguem MQCIH

Especifica o nome do formato IBM MQ dos dados que seguem a estrutura MQCIH.

Na chamada MQPUT ou MQPUT1 , o aplicativo deve configurar esse campo para o valor apropriado aos dados. As regras para codificação desse campo são as mesmas que aquelas para o campo *MDFMT* no MQMD.

Esse nome do formato também será usado para a mensagem de resposta, se o campo *CIRFM* tiver o valor FMNONE

- Para solicitações de DPL, *CIFMT* deve ser o nome do formato da COMMAREA
- Para solicitações 3270, *CIFMT* deve ser CSQCBDCIe *CIRFM* deve ser CSQCBDCO.

As saídas de conversão de dados para esses formatos devem ser instaladas no gerenciador de filas no qual elas devem ser executadas

Se a mensagem de solicitação resultar na geração da mensagem de resposta de erro, a mensagem de resposta de erro terá um nome de formato FMSTR.

Este é um campo de solicitação O comprimento deste campo é fornecido pelo LNFMT O valor inicial deste campo é FMNONE.

**CIFNC (sequência de caracteres de 4 bytes).**

IBM MQ nome da chamada ou CICS função EIBFN.

O valor retornado nesse campo é dependente de *CIRET* ; Consulte [Tabela 690 na página 1070](#). Os seguintes valores são possíveis quando *CIFNC* contém um nome de chamada IBM MQ :

**CFCONN**

Chamada de MQCONN

**CFGET**

Chamada MQGET.

**CFINQ**

Chamada de MQINQ

**CFOPEN**

chamada MQOPEN.

**CFPUT**

chamada MQPUT.

**CFPUT1**

Chamada MQPUT1 .

**CFNONE**

Nenhuma chamada.

Este é um campo de resposta. O comprimento deste campo é fornecido por LNFUNC. O valor inicial deste campo é CFNONE.

**CIGWI (número inteiro assinado de 10 dígitos)**

Intervalo de espera para chamada MQGET emitida pela tarefa de ponte.

Este campo é aplicável apenas quando *CIUOW* tem o valor CUFRST.. Ele permite que o aplicativo de envio especifique o tempo aproximado em milissegundos que as chamadas MQGET emitidas pela ponte devem aguardar pela segunda e pelas mensagens de solicitação subsequentes para a unidade de trabalho iniciada por essa mensagem Isso substitui o intervalo de espera padrão usado pela Ponte.. Podem ser utilizados os seguintes valores especiais:

**WIDFLT**

Intervalo de espera padrão.

Isso faz o CICS bridge aguardar o período especificado quando a ponte foi iniciada.

**WIULIM**

Intervalo de espera ilimitado.

Este é um campo de solicitação O valor inicial deste campo é WIDFLT.

**CIII (número inteiro assinado de 10 dígitos)**

Reservado.

Este é um campo reservado O valor deve ser 0. Esse campo não estará presente se *CIVER* for menor que *CIVER2*.

**CILEN (número inteiro assinado de 10 dígitos)**

Comprimento da estrutura do MQCIH

O valor deve ser um dos seguintes:

**CILEN1**

Comprimento da estrutura do cabeçalho de informações version-1 CICS .

**CILEN2**

Comprimento da estrutura do cabeçalho de informações version-2 CICS .

A constante a seguir especifica o comprimento da versão atual:

**CILENC**

Comprimento da versão atual da estrutura do cabeçalho de informações do CICS

Este é um campo de solicitação O valor inicial desse campo é CILEN2.

### **CILT (número inteiro assinado de 10 dígitos)**

Tipo de link..

Isso indica o tipo de objeto que a ponte deve tentar vincular. O valor deve ser um dos seguintes:

#### **LTPROG**

Programa DPL.

#### **LTTRAN**

Transação 3270..

Este é um campo de solicitação O valor inicial deste campo é LTPROG.

### **CINTI (cadeia de caracteres de 4 bytes).**

Próxima transação para anexar.

Este é o nome da próxima transação retornada pela transação do usuário (geralmente por EXEC CICS RETURN TRANSID). Se não houver nenhuma próxima transação, esse campo será configurado como espaços em branco

Este é um campo de resposta utilizado apenas para transações 3270 O comprimento deste campo é fornecido por LNTRID.. O valor inicial deste campo é 4 espaços em branco.

### **CIODL (número inteiro assinado de 10 dígitos).**

Comprimento de dados de COMMAREA de Saída

Este é o comprimento dos dados do usuário a serem retornados para o cliente em uma mensagem de resposta Esse comprimento inclui o nome do programa de 8 bytes O comprimento do COMMAREA passado para o programa vinculado é o máximo deste campo e o comprimento dos dados do usuário na mensagem de solicitação, menos 8.

**Nota:** O comprimento dos dados do usuário em uma mensagem é o comprimento da mensagem *excluindo* a estrutura MQCIH..

Se o comprimento dos dados do usuário na mensagem de solicitação for menor que *CIODL*, a opção DATALENGTH do comando LINK será usada; isso permite que o LINK seja enviado com eficiência para outra região CICS .

O seguinte valor especial pode ser usado:

#### **OLINPT**

O comprimento de saída é igual ao comprimento de entrada.

Esse valor pode ser necessário mesmo se nenhuma resposta for solicitada, para assegurar que a COMMAREA transmitida para o programa vinculado seja de tamanho suficiente.

Este é um campo de pedido utilizado apenas para programas DPL O valor inicial desse campo OLINPT.

### **CIREA (número inteiro assinado de 10 dígitos)**

IBM MQ código de razão ou feedback ou CICS EIBRESP2.

O valor retornado nesse campo é dependente de *CIRET* ; Consulte [Tabela 690 na página 1070](#).

Este é um campo de resposta. O valor inicial deste campo é RCNONE..

### **CIRET (número inteiro assinado de 10 dígitos)**

Código de retorno da ponte.

Este é o código de retorno do CICS bridge que descreve o resultado do processamento executado pela ponte Os campos *CIFNC*, *CICC*, *CIREA* e *CIAC* podem conter informações adicionais (consulte [Tabela 690 na página 1070](#) .) O valor é um dos seguintes:

#### **CRC000**

(0, X'000 ') Nenhum erro.

#### **CRC001**

(1, X'001 ') instrução EXEC CICS detectou um erro.

**CRC002**

(2, X'002 ') IBM MQ detectou um erro.

**CRC003**

(3, X'003 ') CICS bridge detectou um erro.

**CRC004**

(4, X'004 ') CICS bridge terminou de forma anormal.

**CRC005**

(5, X'005 ') O aplicativo terminou de forma anormal.

**CRC006**

(6, X'006 ') Ocorreu um erro de segurança

**CRC007**

(7, X'007 ') Programa não disponível.

**CRC008**

(8, X'008 ') Segunda ou posterior mensagem na unidade de trabalho atual não recebida dentro do tempo especificado.

**CRC009**

(9, X'009 ') Transação não disponível.

Este é um campo de resposta. O valor inicial desse campo é CRC000.

**CIRFM (sequência de caracteres de 8 bytes).**

IBM MQ nome do formato da mensagem de resposta..

Este é o nome do formato IBM MQ da mensagem de resposta que será enviada em resposta à mensagem atual As regras para codificar isso são as mesmas para o campo *MDFMT* no MQMD.

Este é um campo de pedido utilizado apenas para programas DPL O comprimento deste campo é fornecido pelo LNFMT O valor inicial deste campo é FMNONE.

**CIRSI (sequência de caracteres de 4 bytes)**

Reservado.

Este é um campo reservado O valor deve ser 4 espaços em branco. O comprimento deste campo é fornecido pelo LNRSID.

**CIRS1 (sequência de caracteres de 8 bytes).**

Reservado.

Este é um campo reservado O valor deve ser 8 espaços em branco.

**CIRS2 (sequência de caracteres de 8 bytes).**

Reservado.

Este é um campo reservado O valor deve ser 8 espaços em branco.

**CIRS3 (sequência de caractere de 8 bytes)**

Reservado.

Este é um campo reservado O valor deve ser 8 espaços em branco.

**CIRS4 (número inteiro assinado de 10 dígitos)**

Reservado.

Este é um campo reservado O valor deve ser 0. Esse campo não estará presente se *CIVER* for menor que *CIVER2*.

**CIRTI (cadeia de caracteres de 4 bytes)**

Reservado.



Este é um campo reservado O valor deve ser 4 espaços em branco. O comprimento deste campo é fornecido por LNTRID..

### **CISC (sequência de caracteres de 4 bytes)**

Código inicial da transação.

Este é um indicador que especifica se a ponte emula uma transação de terminal ou uma transação START (START). O valor deve ser um dos seguintes:

#### **SCSTRT**

Iniciar.

#### **SCDATA**

Dados de início

#### **SCTERM**

Finalizar entrada.

#### **SCNONE**

Nenhum.

Na resposta da ponte, esse campo é configurado com o código de início apropriado para o próximo ID de transação contido no campo *CINTI* Os seguintes códigos de início são possíveis na resposta:

- SCSTRT
- SCDATA
- SCTERM

Para CICS Transaction Server 1.2 , esse campo é apenas um campo de solicitação; seu valor na resposta é indefinido..

Para CICS Transaction Server 1.3 e liberações subsequentes, este é um campo de solicitação e de resposta.

Este campo é utilizado apenas para transações 3270 O comprimento deste campo é fornecido por LNSTCO... O valor inicial desse campo é SCNONE.

### **CISID (sequência de caracteres de 4 bytes).**

Identificador de estruturação.

O valor deve ser:.

#### **CISIDV**

Identificador para a estrutura do cabeçalho de informações CICS .

Este é um campo de solicitação O valor inicial desse campo é CISIDV.

### **CITES (número inteiro assinado de 10 dígitos)**

Status no término da tarefa

Este campo mostra o status da transação do usuário no término da tarefa Um dos seguintes valores é retornado:

#### **TENOSY**

Não sincronizado..

A transação do usuário ainda não foi concluída e não possui sincronização. Nesse caso, o campo *MDMT* no MQMD é MTRQST

#### **TECMIT**

Confirmar unidade de trabalho.

A transação do usuário ainda não foi concluída, mas sincroniza a primeira unidade de trabalho. Nesse caso, o campo *MDMT* no MQMD é MTDGRM

#### **TEBACK**

Unidade de trabalho de volta.

A transação do usuário ainda não foi concluída A unidade de trabalho atual será restaurada..  
Nesse caso, o campo *MDMT* no MQMD é MTDGRM

#### **TEENDT**

Terminar tarefa.

A transação do usuário foi encerrada (ou encerrada de forma anormal). Nesse caso, o campo *MDMT* no MQMD é MTRPLY.

Este é um campo de resposta utilizado apenas para transações 3270 O valor inicial deste campo é TENOSY.

#### **CITI (sequência de caracteres de 4 bytes)**

Transação a ser anexada

Se *CILT* tiver o valor LTTRAN, *CITI* será o identificador de transação da transação do usuário a ser executada; um valor não em branco deve ser especificado nesse caso.

Se *CILT* tiver o valor LTPROG, *CITI* será o código de transação sob o qual todos os programas na unidade de trabalho devem ser executados. Se o valor especificado estiver em branco, o código de transação padrão da ponte DPL (CKBP) do CICS será usado Se o valor não estiver em branco, ele deverá ter sido definido como CICS como uma TRANSACTION local com um programa inicial de CSQCBP00... Este campo é aplicável apenas quando *CIUOW* possui o valor CUFRST ou CUONLY.

Este é um campo de solicitação O comprimento deste campo é fornecido por LNTRID.. O valor inicial deste campo é 4 espaços em branco.

#### **CIUOW (número inteiro assinado de 10 dígitos)**

Controle da unidade de trabalho

Isso controla o processamento de unidade de trabalho executado pelo CICS bridge É possível solicitar que a ponte execute uma única transação ou um ou mais programas em uma unidade de trabalho. O campo indica se o CICS bridge deve iniciar uma unidade de trabalho, executar a função solicitada dentro da unidade de trabalho atual ou terminar a unidade de trabalho confirmando-a ou fazendo backup dela. Várias combinações são suportadas, para otimizar os fluxos de transmissão de dados

O valor deve ser um dos seguintes:

#### **CUONLY**

Inicie a unidade de trabalho, execute a função e, em seguida, confirme a unidade de trabalho (DPL e 3270).

#### **CUCONT**

Dados adicionais para a unidade de trabalho atual (3270 apenas)

#### **CUFRST**

Iniciar unidade de trabalho e executar função (somente DPL).

#### **CUMIDL**

Executar função na unidade de trabalho atual (somente DPL).

#### **CULAST**

Execute a função e, em seguida, confirme a unidade de trabalho (somente DPL)

#### **CUCMIT**

Confirmar a unidade de trabalho (somente DPL).

#### **CUBACK**

Voltar a unidade de trabalho (somente DPL).

Este é um campo de solicitação O valor inicial deste campo é CUONLY.

#### **CIVER (número inteiro assinado de 10 dígitos)**

Número de versão da estrutura.

O valor deve ser um dos seguintes:

**CIVER1**

Version-1 CICS estrutura do cabeçalho de informações.

**CIVER2**

Version-2 CICS estrutura do cabeçalho de informações.

Os campos que existem apenas na versão mais recente da estrutura são identificados como tal nas descrições dos campos A constante a seguir especifica o número da versão atual:

**CIVERC**

Versão atual da estrutura do cabeçalho de informações CICS .

Este é um campo de solicitação O valor inicial desse campo é CIVER2.

**Valores iniciais**

<i>Tabela 691. Campos em MQCIH</i>		
<b>Nome de Campo</b>	<b>Nome da constante</b>	<b>Valor da constante</b>
<i>CISID</i>	CISIDV	'CIH→'
<i>CIVER</i>	CIVER2	2
<i>CILEN</i>	CILEN2	180
<i>CIENC</i>	Nenhum	0
<i>CICSI</i>	Nenhum	0
<i>CIFMT</i>	FMNONE	Espaços em branco
<i>CIFLG</i>	CIFNON	0
<i>CIRET</i>	CRC000	0
<i>CICC</i>	CCOK	0
<i>CIREA</i>	RCNONE	0
<i>CIUOW</i>	CUONLY	273
<i>CIGWI</i>	WIDFLT	-2
<i>CILT</i>	LTPROG	1
<i>CIODL</i>	OLINPT	-1
<i>CIFKT</i>	Nenhum	0
<i>CIADS</i>	ADNONE	0
<i>CICT</i>	CTNO	0
<i>CITES</i>	TENOSY	0
<i>CIFAC</i>	FCNONE	Nulos
<i>CIFNC</i>	CFNONE	Espaços em branco
<i>CIAC</i>	Nenhum	Espaços em branco
<i>CIAUT</i>	Nenhum	Espaços em branco
<i>CIRS1</i>	Nenhum	Espaços em branco
<i>CIRFM</i>	FMNONE	Espaços em branco
<i>CIRSI</i>	Nenhum	Espaços em branco
<i>CIRTI</i>	Nenhum	Espaços em branco

Tabela 691. Campos em MQCIH (continuação)

Nome de Campo	Nome da constante	Valor da constante
CITI	Nenhum	Espaços em branco
CIFL	Nenhum	Espaços em branco
CIAI	Nenhum	Espaços em branco
CISC	SCNONE	Espaços em branco
CICNC	Nenhum	Espaços em branco
CINTI	Nenhum	Espaços em branco
CIRS2	Nenhum	Espaços em branco
CIRS3	Nenhum	Espaços em branco
CICP	Nenhum	0
CIEO	Nenhum	0
CIII	Nenhum	0
CIRS4	Nenhum	0

**Notas:**

1. O símbolo – representa um único caractere em branco.

**Declaração RPG**

```

D* .1.....2.....3.....4.....5.....6.....7..
D* MQCIH Structure
D*
D* Structure identifier
D  CISID          1      4  INZ('CIH ')
D* Structure version number
D  CIVER          5      8I 0 INZ(2)
D* Length of MQCIH structure
D  CILEN          9     12I 0 INZ(180)
D* Reserved
D  CIENC         13     16I 0 INZ(0)
D* Reserved
D  CICSI         17     20I 0 INZ(0)
D* MQ format name of data that followsMQCIH
D  CIFMT         21     28  INZ(' ')
D* Flags
D  CIFLG         29     32I 0 INZ(0)
D* Return code from bridge
D  CIRET         33     36I 0 INZ(0)
D* MQ completion code or CICSEIBRESP
D  CICC          37     40I 0 INZ(0)
D* MQ reason or feedback code, or CICSEIBRESP2
D  CIREA         41     44I 0 INZ(0)
D* Unit-of-work control
D  CIUOW         45     48I 0 INZ(273)
D* Wait interval for MQGET call issuedby bridge task
D  CIGWI         49     52I 0 INZ(-2)
D* Link type
D  CILT          53     56I 0 INZ(1)
D* Output COMMAREA data length
D  CIODL         57     60I 0 INZ(-1)
D* Bridge facility release time
D  CIFKT         61     64I 0 INZ(0)
D* Send/receive ADS descriptor
D  CIADS         65     68I 0 INZ(0)
D* Whether task can beconversational
D  CICT          69     72I 0 INZ(0)
D* Status at end of task
D  CITES         73     76I 0 INZ(0)
D* Bridge facility token

```

```

D CIFAC          77      84  INZ(X'0000000000000000-
D
D* MQ call name or CICS EIBFNfunction
D CIFNC          85      88  INZ(' ')
D* Abend code
D CIAC           89      92  INZ
D* Password or passticket
D CIAUT          93     100  INZ
D* Reserved
D CIRS1          101     108  INZ
D* MQ format name of reply message
D CIRFM          109     116  INZ(' ')
D* Remote CICS system ID to use
D CIRSI          117     120  INZ
D* CICS RTRANSID to use
D CIRTI          121     124  INZ
D* Transaction to attach
D CITI           125     128  INZ
D* Terminal emulated attributes
D CIFL           129     132  INZ
D* AID key
D CIAI           133     136  INZ
D* Transaction start code
D CISC           137     140  INZ(' ')
D* Abend transaction code
D CICNC          141     144  INZ
D* Next transaction to attach
D CINTI          145     148  INZ
D* Reserved
D CIRS2          149     156  INZ
D* Reserved
D CIRS3          157     164  INZ
D* Cursor position
D CICP           165     168I 0 INZ(0)
D* Offset of error in message
D CIEO           169     172I 0 INZ(0)
D* Reserved
D CIII           173     176I 0 INZ(0)
D* Reserved
D CIRS4          177     180I 0 INZ(0)
D*

```



## **MQCMHO (Criar opções de manipulação de mensagens) no IBM i**

A estrutura **MQCMHO** permite que os aplicativos especifiquem opções que controlam como as manipulações de mensagens são criadas

### **Visão Geral**

#### **Propósito**

A estrutura é um parâmetro de entrada na chamada **MQCRTMH ..**

#### **Conjunto de caracteres e codificação**

Os dados em **MQCMHO** devem estar no conjunto de caracteres do aplicativo e na codificação do aplicativo (ENNAT)..

- [“Campos” na página 1081](#)
- [“Valores iniciais” na página 1083](#)
- [“Declaração RPG” na página 1083](#)

### **Campos**

A estrutura **MQCMHO** contém os seguintes campos; os campos são descritos em ordem alfabética:

#### **CMOPT (número inteiro assinado de 10 dígitos)**

Uma das seguintes opções pode ser especificada:

##### **CMVAL**

Quando **MQSETMP** é chamado para configurar uma propriedade nessa manipulação de mensagem, o nome da propriedade é validado para assegurar que:

- Não contém caracteres inválidos.
- não inicia "JMS" ou "usr.JMS", exceto para o seguinte:
  - JMSCorrelationID
  - JMSReplyTo
  - JMSType
  - JMSXGroupID
  - JMSXGroupSeq

Esses nomes são reservados para propriedades JMS .

- não é uma das seguintes palavras-chave, em qualquer mistura de letras maiúsculas ou minúsculas:
  - "E"
  - "ENTRE"
  - "ESCAPE"
  - "FALSE"
  - "IN"
  - "IS"
  - "gosto"
  - "não"
  - "NULL"
  - "OU"
  - "TRUE"
- não começa com "Corpo". ou "Raiz." (exceto para "Root.MQMD.").

Se a propriedade for definida pelo MQ("mq. \*") e o nome for reconhecido, os campos do descritor de propriedades serão configurados para os valores corretos para a propriedade. Se a propriedade não for reconhecida, o campo *Support* do descritor de propriedade será configurado como **PDSUPO** (para obter mais informações consulte [PDSUP](#)).

### **CMDEFV**

Isso especifica que o nível padrão de validação de nomes de propriedades ocorre

O nível padrão de validação é equivalente ao especificado por **CMVAL**

Em uma liberação futura, uma opção administrativa poderá ser definida, o que alterará o nível de validação que ocorrerá quando **CMDEFV** for definido

Esse é o valor-padrão.

### **CMNOVA**

Não ocorre nenhuma validação no nome da propriedade. Consulte a descrição de **CMVAL**

**Opção padrão:** Se nenhuma das opções descritas anteriormente nesta seção for necessária, a seguinte opção poderá ser usada:

### **CMNONE**

Todas as opções assumem seus valores padrão. Utilize este valor para indicar que nenhuma outra opção foi especificada.. O **CMNONE** auxilia a documentação do programa; não é desejado que essa opção seja usada com nenhuma outra, mas como seu valor é zero, tal uso não pode ser detectado.

Isto é sempre um campo de entrada. O valor inicial desse campo é **CMDEFV**.

### **CMSID (número inteiro assinado de 10 dígitos)**

Este é o identificador de estrutura; o valor deve ser:

**CMSIDV**

Identificador para criar estrutura de opções de manipulação de mensagem.

Isto é sempre um campo de entrada. O valor inicial desse campo é **CMSIDV**.

**CMVER (número inteiro assinado de 10 dígitos)**

Este é o número da versão da estrutura; o valor deve ser:

**CMVER1**

Version-1 criar estrutura de opções de manipulação de mensagens.

A constante a seguir especifica o número da versão atual:

**CMVERC**

A versão atual da estrutura de opções de manipulação de mensagem de criação

Isto é sempre um campo de entrada. O valor inicial desse campo é **CMVER1**.

**Valores iniciais**

Tabela 692. Campos em MQCMHO		
Nome de Campo	Nome da constante	Valor da constante
CMSID	CMSIDV	'CMHO'
CMVER	CMVER1	1
CMOPT	CMDEFV	0

**Declaração RPG**

```
D* MQCMHO Structure
D*
D*
D* Structure identifier
D CMSID          1      4    INZ('CMHO')
D*
D* Structure version number
D CMVER          5      8I 0 INZ(1)
D*
D* Options that control the action of MQCRTMH
D CMOPT          9     12I 0 INZ(0)
```

**IBM i MQCNO (Opções de Conexão) no IBM i**

A estrutura MQCNO permite que o aplicativo especifique opções relacionadas à conexão ao gerenciador de fila local.

**Visão Geral**

**Propósito:** A estrutura é um parâmetro de entrada / saída na chamada MQCONNX.

**Versão:** A versão atual do MQCNO é CNVER6. Os campos que existem apenas nas versões mais recentes da estrutura são identificados como tal nas descrições a seguir:

O arquivo COPY fornecido contém a versão mais recente de MQCNO que é suportada pelo ambiente, mas com o valor inicial do campo CNVER configurado como CNVER1. Para usar campos que não estão

presentes na estrutura version-1 , o aplicativo deve configurar o campo CNVER para o número da versão necessária.

**Codificação e conjunto de caracteres:** os dados no MQCNO devem estar no conjunto de caracteres fornecido pelo atributo do gerenciador de filas **CodedCharSetId** e a codificação do gerenciador de filas locais fornecido pelo ENNAT.

- [“Campos” na página 1084](#)
- [“Valores iniciais” na página 1089](#)
- [“Declaração RPG” na página 1090](#)

## Campos

A estrutura MQCNO contém os seguintes campos; os campos são descritos em **ordem alfabética**:

### CCDTUL (número inteiro assinado de 10 dígitos)

CCDTUL é o comprimento da sequência identificada por CCDTUP ou CCDTUO que contém uma URL que identifica o local da tabela de canal de conexão do cliente a ser usado para a conexão...

Use CCDTUL apenas quando o aplicativo que emite a chamada MQCONNX estiver em execução como um IBM MQ MQI client.

Esta é uma alternativa programática para configurar as variáveis de ambiente [MQCHLLIB](#) e [MQCHLTA](#) .

Se o aplicativo não estiver executando como um cliente, CCDTUL será ignorado.

Esse campo será ignorado se o CNVER for menor que CNVER6

### CCDTUO (número inteiro assinado de 10 dígitos)

CCDTUO é o deslocamento em bytes, a partir do início da estrutura MQCNO, para uma sequência que contém uma URL que identifica o local da tabela do canal de conexão do cliente para usar para a conexão. O deslocamento pode ser positivo ou negativo

Use CCDTUL apenas quando o aplicativo que emite a chamada MQCONNX estiver em execução como um IBM MQ MQI client.

**Importante:** É possível usar apenas um de CCDTUP e CCDTUO A chamada falha com o código de razão RC2600 se ambos os campos forem diferentes de zero.

Esta é uma alternativa programática para configurar as variáveis de ambiente [MQCHLLIB](#) e [MQCHLTA](#) .

Se o aplicativo não estiver executando como um cliente, CCDTUO será ignorado.

Esse campo será ignorado se o CNVER for menor que CNVER6

### CCDTUP (ponteiro)

CCDTUP é um ponteiro opcional para uma sequência que contém uma URL, para identificar o local da tabela do canal de conexão do cliente para usar para a conexão.

Use CCDTUP apenas quando o aplicativo que está emitindo a chamada MQCONNX estiver em execução como um IBM MQ MQI client.

**Importante:** É possível usar apenas um de CCDTUP e CCDTUO A chamada falha com o código de razão RC2600 se ambos os campos forem diferentes de zero.

Esta é uma alternativa programática para configurar as variáveis de ambiente [MQCHLLIB](#) e [MQCHLTA](#) .

Se o aplicativo não estiver em execução como cliente, CCDTUP será ignorado.

Esse campo será ignorado se o CNVER for menor que CNVER6

### CNAN (cadeia de caracteres de 28 bytes)

O nome configurado pelo aplicativo para identificar a conexão com o gerenciador de filas O valor inicial do campo é caracteres nulos.



Esse campo será ignorado se CNVER for menor que CNVER7..

### **CNCCO (número inteiro assinado de 10 dígitos)**

Este é o deslocamento em bytes de uma estrutura de definição de canal MQCD a partir do início da estrutura MQCNO.

### **CNCCP (ponteiro)**

Este é um ponteiro para uma estrutura de definição de canal MQCD.

### **CNCONID (sequência de caracteres de 24 bytes).**

Identificador de conexão exclusivo Esse campo permite que o gerenciador de filas identifique de forma confiável um processo de aplicativo designando a ele um identificador exclusivo quando ele se conecta pela primeira vez ao gerenciador de filas..

Os aplicativos usam o identificador de conexão para propósitos de correlação ao fazer chamadas de PUT e GET. Todas as conexões são designadas a um identificador pelo gerenciador de filas, independentemente de como a conexão foi estabelecida...

É possível usar o identificador de conexão para forçar o término de uma unidade de trabalho de longa execução. Para fazer isso, especifique o identificador de conexão usando o comando PCF 'Stop Connection' ou o comando MQSC STOP CONN. Para obter mais informações sobre como usar esses comandos, consulte os links relacionados..

O valor inicial do campo é de 24 bytes nulos.

### **CNCT (cadeia de bits de 128 bytes)**

Esta é uma tag que o gerenciador de filas associa com os recursos que são afetados pelo aplicativo durante esta conexão

Tag de conexão do gerenciador de filas

Cada aplicativo ou instância do aplicativo deve usar um valor diferente para a tag, para que o gerenciador de filas possa serializar corretamente o acesso aos recursos afetados. Consulte as descrições das opções CN\* CT\* para obter mais detalhes. A tag deixa de ser válida quando o aplicativo é encerrado ou emite a chamada MQDISC.

Use o valor especial a seguir se nenhuma tag for necessária:

#### **CTNONE**

Nenhuma tag de conexão foi especificada

O valor é binário zero para o comprimento do campo.

Este é um campo de entrada. O comprimento deste campo é fornecido pelo LNCTAG O valor inicial deste campo é CTNONE. Esse campo será ignorado se CNVER for menor que CNVER3

Use o campo ConnTag ao conectar a um gerenciador de filas do z/OS .

### **CNNORES2 (sequência de caracteres de 4 bytes)**

Um campo reservado para preencher a estrutura para um limite de 64 bits. O valor inicial do campo é zero binário para o comprimento do campo.

Esse campo será ignorado se CNVER for menor que CNVER7..

### **CNOPT (número inteiro assinado de 10 dígitos)**

Opções que controlam a ação de MQCONN..

#### **Opções de Ligação**

As opções de ligação controlam o tipo de ligação IBM MQ que é usado; especifique apenas uma destas opções:

#### **CNSBND**

Ligação padrão.

A opção de ligação padrão faz com que o aplicativo e o agente do gerenciador de filas locais sejam executados em unidades separadas de execução, geralmente em processos separados. O acordo mantém a integridade do gerenciador de fila; ou seja, ele protege o gerenciador de filas de programas errantes..

Use CNSBND em situações em que o aplicativo pode não ter sido totalmente testado ou pode não ser confiável ou não confiável. CNSBND é o padrão..

CNSBND é definido para a documentação do programa de auxílio. Não utilize esta opção com qualquer outra opção controlando o tipo de ligação utilizado; mas como seu valor é zero, tal uso não pode ser detectado.

Essa opção é suportada em todos os ambientes

### **CNFBND**

Ligação de atalho.

A opção de ligação de atalho faz com que o aplicativo e o agente do gerenciador de filas locais façam parte da mesma unidade de execução. O atalho está em contraste com a ligação padrão, em que o aplicativo e o agente do gerenciador de fila local são executados em unidades separadas de execução.

CNFBND será ignorado se o gerenciador de filas não suportar esse tipo de ligação; o processamento continua como se a opção não tivesse sido especificada

CNFBND pode ser vantajoso em situações em que vários processos consomem mais recursos do que o recurso geral usado pelo aplicativo. Um aplicativo que usa a ligação de atalho é conhecido como um *aplicativo confiável*

Considere os pontos importantes a seguir ao decidir se deve usar a ligação de atalho:

- **A utilização da opção CNFBND não evita que um aplicativo altere ou corrompa mensagens e outras áreas de dados pertencentes ao gerenciador de filas. Use essa opção somente em situações em que você tenha avaliado totalmente esses problemas,**
- O aplicativo não deve usar sinais assíncronos ou interrupções de cronômetro (como sigkill) com CNFBND. Há também restrições no uso de segmentos de memória compartilhada.
- O aplicativo não deve ter mais de um encadeamento conectado ao gerenciador de filas por vez.
- O aplicativo deve usar a chamada MQDISC para desconectar do gerenciador de filas.
- O aplicativo deve ser concluído antes de terminar o gerenciador de filas com o comando endmqm

Os seguintes pontos se aplicam ao uso de CNFBND nos ambientes indicados:

- No IBM i, a tarefa deve ser executada sob o perfil do usuário QMQM que pertence ao grupo QMQMADM. Além disso, o programa não deve terminar de forma anormal, caso contrário, poderão ocorrer resultados imprevisíveis.

Para obter mais informações sobre as implicações do uso de aplicativos confiáveis, consulte [Conectando a um gerenciador de filas usando a chamada MQCONNX](#) e [Restrições para aplicativos confiáveis](#).

### **CNSHBD**

Ligações Compartilhadas.

A opção de ligações compartilhadas faz o aplicativo e o agente do gerenciador de filas locais serem executados em unidades separadas de execução, geralmente em processos separados. O acordo mantém a integridade do gerenciador de fila; ou seja, ele protege o gerenciador de filas de programas errantes.. No entanto, alguns recursos são compartilhados entre o aplicativo e o agente do gerenciador de filas locais. CNSHBD será ignorado se o gerenciador de fila não suportar esse tipo de ligação.. O processamento continuará, embora a opção não tenha sido especificada.

## **CNIBND**

### Ligações Isoladas

A opção de ligações isoladas faz com que o aplicativo e o agente do gerenciador de filas locais sejam executados em unidades separadas de execução, geralmente em processos separados. O acordo mantém a integridade do gerenciador de fila; ou seja, ele protege o gerenciador de filas de programas errantes.. O processo de aplicativo e o agente do gerenciador de fila local são isolados uns dos outros, pois não compartilham recursos CNIBND será ignorado se o gerenciador de filas não suportar esse tipo de ligação O processamento continuará, embora a opção não tenha sido especificada.

## **Opções de compartilhamento de manipulação**

As seguintes opções controlam o compartilhamento de identificadores entre diferentes encadeamentos (unidades de processamento paralelo) dentro do mesmo processo. Apenas uma dessas opções pode ser especificada..

### **CNHSN**

Nenhum compartilhamento de identificador entre os encadeamentos

A opção sem compartilhamento de identificador entre encadeamentos indica que os identificadores de conexão e de objeto podem ser usados apenas pelo encadeamento que fez com que o identificador fosse alocado; ou seja, o encadeamento que emitiu a chamada MQCONN, MQCONNX ou MQOPEN Os identificadores não podem ser usados por outros encadeamentos pertencentes ao mesmo processo

### **CNHSB**

Compartilhamento de alça serial entre encadeamentos, com bloqueio de chamada

O compartilhamento de identificador serial entre encadeamentos, com bloqueio de chamada, opção indica que os identificadores de conexão e de objeto alocados por um encadeamento de um processo podem ser usados por outros encadeamentos pertencentes ao mesmo processo. No entanto, apenas um encadeamento por vez pode usar qualquer identificador específico, ou seja, apenas o uso serial de um identificador é permitido. Se um encadeamento tentar usar um identificador que já esteja em uso por outro encadeamento, os blocos de chamada (espera) até que o identificador se torne disponível.

### **CNHSNB**

Compartilhamento de identificador de série entre encadeamentos, sem bloqueio de chamada

O compartilhamento de identificador serial entre encadeamentos, sem bloqueio de chamada, opção é o mesmo que o " *com a opção blocking* ", exceto que, se a manipulação estiver em uso por outro encadeamento, a chamada será concluída imediatamente com CCFAIL e RC2219 em vez de bloquear até a manipulação se tornar disponível.

Um encadeamento pode ter zero ou um identificador não compartilhado, mais zero ou mais identificadores compartilhados:

- Cada chamada MQCONN ou MQCONNX que especifica CNHSN retorna um novo identificador não compartilhado na primeira chamada e o mesmo identificador não compartilhado em chamadas subsequentes (assumindo nenhuma chamada MQDISC interveniente). O código de razão é RC2002 para a segunda e as chamadas mais recentes
- Cada chamada MQCONNX que especifica CNHSB ou CNHSNB retorna um novo identificador compartilhado em cada chamada.

Os identificadores de objeto herdam as mesmas propriedades de compartilhamento que o identificador de conexão especificado na chamada MQOPEN que criou o identificador de objeto Além disso, as unidades de trabalho herdam as mesmas propriedades de compartilhamento que o identificador de conexão usado para iniciar a unidade de trabalho; se a unidade de trabalho for iniciada em um encadeamento usando um identificador compartilhado, a unidade de trabalho poderá ser atualizada em outro encadeamento usando o mesmo identificador

Se você não especificar uma opção de compartilhamento de manipulação, o padrão será determinado pelo ambiente:

- No ambiente do Microsoft Transaction Server (MTS), o padrão é o mesmo que CNHSB.
- Em outros ambientes, o padrão é igual a CNHSN.

### **Opções de reconexão.**

Opções de reconexão determinam se uma conexão é reconectável. Apenas conexões do cliente são reconectáveis.

#### **CNRCDF**

A opção de reconexão é resolvida para seu valor padrão. Se nenhum padrão for configurado, o valor dessa opção será resolvido para DISABLED. O valor da opção é passado para o servidor e pode ser consultado por **PCF** e **MQSC**.

#### **CNRC**

O aplicativo pode ser reconectado a qualquer gerenciador de filas consistente com o valor do parâmetro MQCONNX **QMNAME**. Use a opção CNRC somente se não houver nenhuma afinidade entre o aplicativo cliente e o gerenciador de fila com o qual ele estabeleceu inicialmente uma conexão. O valor da opção é passado para o servidor e pode ser consultado por **PCF** e **MQSC**.

#### **CNRCDe**

O aplicativo não pode ser reconectado.. O valor da opção não é transmitido ao servidor.

#### **CNRCQM**

O aplicativo só pode ser reconectado ao gerenciador de filas com o qual ele se conectou originalmente. Use esse valor se um cliente puder ser reconectado, mas houver uma afinidade entre o aplicativo cliente, e o gerenciador de filas com o qual ele originalmente estabeleceu uma conexão. Escolha este valor se você deseja que um cliente restabeleça automaticamente a conexão com a instância em espera de um gerenciador de filas altamente disponível. O valor da opção é passado para o servidor e pode ser consultado por **PCF** e **MQSC**.

Use as opções CNRC, CNRCDe CNRCQM apenas para conexões do cliente. Se as opções forem usadas para uma conexão de ligação, o MQCONNX falhará com o código de conclusão, MQCC\_FAILED e o código de razão, MQRC\_OPTIONS\_ERROR

**Opção padrão:** se nenhuma das opções descritas for necessária, a opção a seguir poderá ser usada:

#### **CNNONE**

Nenhuma opção foi especificada

CNNONE é definido para a documentação do programa de auxílio. Não se pretende que essa opção seja usada com qualquer outra opção CN\*, mas como seu valor é zero, tal uso não pode ser detectado.

### **CNSCO (número inteiro assinado de 10 dígitos)**

Este é o deslocamento em bytes de uma estrutura MQSCO do início da estrutura MQCNO.

Esse campo será ignorado se CNVER for menor que CNVER4

### **CNSCP (ponteiro)**

Este é o endereço de uma estrutura MQSCO..

Esse campo será ignorado se CNVER for menor que CNVER4

### **CNSECPO (número inteiro assinado de 10 dígitos)**

Compensação de parâmetros de segurança. O deslocamento da estrutura MQCSP usado para especificar um ID do usuário e senha.

O valor pode ser positivo ou negativo. O valor inicial desse campo é 0.

Esse campo será ignorado, se CNVER for menor que CNVER5

### **CNSECPP (ponteiro)**

Ponteiro de parâmetros de segurança Endereço da estrutura MQCSP usada para especificar um ID do usuário e uma senha...

O valor inicial desse campo é um ponteiro nulo ou bytes nulos

Esse campo será ignorado, se CNVER for menor que CNVER5

### **CNSID (cadeia de caracteres de 4 bytes).**

O identificador de estrutura para a estrutura MQCNO.

O valor deve ser:.

#### **CNSIDV**

Identificador para a estrutura de opções de conexão

Isto é sempre um campo de entrada. O valor inicial desse campo é CNSIDV.

### **CNVER (número inteiro assinado de 10 dígitos)**

O número da versão da estrutura para a estrutura MQCNO..

O valor deve ser:.

#### **CNVER6**

Estrutura de opções de conexão Version-6 .

Esta versão é suportada em todos os ambientes

#### **CNVER7**

Estrutura de opções de conexão Version-7 .

Esta versão é suportada em todos os ambientes

A constante a seguir especifica o número da versão atual:

#### **CNVERC**

A versão atual da estrutura de opções de conexão.

Isto é sempre um campo de entrada. O valor inicial desse campo é CNVER7.

## **Valores iniciais**

*Tabela 693. Campos em MQCNO*

<b>Nome de Campo</b>	<b>Nome da constante</b>	<b>Valor da constante</b>
CNSID	CNSIDV	' CNO~
CNVER	CNVER5	1
CNOPT	CNNONE	0
CNCCO	Nenhum	0
CNCCP	Nenhum	Ponteiro nulo ou bytes nulos
CNCT	CTNONE	Nulos
CNSCP	Nenhum	Ponteiro nulo ou bytes nulos
CNSCO	Nenhum	0
CNCONID	Nenhum	Nulos
CNSECPO	Nenhum	0
CNSECPP	Nenhum	Ponteiro nulo ou bytes nulos

Tabela 693. Campos em MQCNO (continuação)

Nome de Campo	Nome da constante	Valor da constante
CCDTUL	Nenhum	0
CCDTUO	Nenhum	0
CCDTUP	Nenhum	Ponteiro nulo ou bytes nulos

**Notas:**

1. O símbolo - representa um único caractere em branco.

**Declaração RPG**

```

D*****
D**
D**          IBM MQ for IBM i          **
D**          **
D** FILE NAME:      CMQCNOG           **
D**          **
D** DESCRIPTION:   MQCNO Structure -- Connect Options **
D**          **
D*****
D** <N_OCO_COPYRIGHT>                **
D** Licensed Materials - Property of IBM **
D**          **
D** 5724-H72                          **
D** (c) Copyright IBM Corp. 1993, 2024. All Rights Reserved. **
D**          **
D** US Government Users Restricted Rights - Use, duplication or **
D** disclosure restricted by GSA ADP Schedule Contract with **
D** IBM Corp.                          **
D** <NOC_COPYRIGHT>                  **
D*****
D** FUNCTION:      This file declares the structure MQCNO, **
D**                which is used by the main MQI.          **
D**          **
D** PROCESSOR:    RPG (ILE)           **
D**          **
D*****
D*
D*
D*****
D** <BEGIN_BUILDINFO>                **
D** Generated on:  08/02/16 13:50     **
D** Build Level:   L000000           **
D** Build Type:    Production         **
D** Pointer Size:  128 Bit           **
D** Source File:   **
D** CMQCNOG       **
D** <END_BUILDINFO>                  **
D*****
D*
D*
D* ..1....:....2....:....3....:....4....:....5....:....6....:....7..
D*
D*
D* MQCNO Structure
D*
D* Structure identifier
D  CNSID          1      4  INZ('CNO ')
D* Structure version number
D  CNVER          5      8I 0 INZ(1)
D* Options that control the action of MQCONN
D  CNOPT          9      12I 0 INZ(0)
D* Ver:1 **
D* Offset of MQCD structure for client connection
D  CNCCO         13      16I 0 INZ(0)
D* Address of MQCD structure for client connection
D  CNCCP         17      32*  INZ(*NULL)
D* Ver:2 **
D* Queue managerconnection tag
D  CNCT          33      160  INZ(X'0000000000000000-
D                                0000000000000000000000-
D                                0000000000000000000000-

```

```

D          00000000000000000000000000000000-
D          00000000000000000000000000000000-
D          00000000000000000000000000000000-
D          00000000000000000000000000000000-
D          00000000000000000000000000000000-
D          00000000000000000000000000000000-
D          00000000000000000000000000000000-
D          00000000000000000000000000000000-
D          00000000000000000000000000000000-
D* Ver:3 **
D* Address of MQSCO structure for client connection
D CNSCP          161      176*  INZ(*NULL)
D* Offset of MQSCO structure for client connection
D CNSCO          177      180I 0 INZ(0)
D* Ver:4 **
D* Unique Connection Identifier
D CNCONID        181      204      INZ(X'000000000000000000-
D          00000000000000000000000000-
D          000000')
D* Offset of MQCSP structure
D CNSECP0        205      208I 0 INZ(0)
D* Address of MQCSP structure
D CNSECPP        209      224*  INZ(*NULL)
D* Ver:5 **
D* Address of CCDT URL string
D CNCCDTUP       225      240*  INZ(*NULL)
D* Offset of CCDT URL string
D CNCCDTUO       241      244I 0 INZ(0)
D* Length of CCDT URL
D CNCCDTUL       245      248I 0 INZ(0)
D* Ver:6 **
D*
D*****
D** End of CMQCN0G          **
D*****

```

## IBM i MQCSP (Parâmetros de segurança) no IBM i

Resumo da estrutura do MQCSP para IBM i

### Visão Geral

**Propósito:** a estrutura MQCSP permite que o serviço de autorização autentique um ID do usuário e senha. Você especifica a estrutura de parâmetros de segurança de conexão do MQCSP em uma chamada MQCONNX.

**Conjunto de caracteres e codificação:** Os dados em MQCSP devem estar no conjunto de caracteres fornecido pelo atributo do gerenciador de filas do **CodedCharSetId** e a codificação do gerenciador de fila local fornecido por ENNAT...

- [“Campos” na página 1091](#)
- [“Valores iniciais” na página 1093](#)
- [“Declaração RPG” na página 1094](#)

### Campos

A estrutura MQCSP contém os campos a seguir; os campos são descritos em **ordem alfabética**:

#### CSAUTHT (número inteiro assinado de 10 dígitos)

Esse é o tipo de autenticação a ser executado

Os valores válidos são:

##### CSAN

Não use campos de ID do usuário e senha.

##### CSAUIAP

Autenticar campos de ID do usuário e senha.

Este é um campo de entrada. O valor inicial desse campo é CSAN..

### **CSCPPL (número inteiro assinado de 10 dígitos)**

Esse é o comprimento da senha a ser usada na autenticação

O comprimento máximo da senha não é dependente da plataforma Se o comprimento da senha for maior do que o permitido, a solicitação de autenticação falhará com um RC2035

Este é um campo de entrada. O valor inicial desse campo é 0.

### **CSCPPO (número inteiro assinado de 10 dígitos)**

Este é o deslocamento em bytes da senha a ser usada na autenticação.

O deslocamento pode ser positivo ou negativo

Este é um campo de entrada. O valor inicial desse campo é 0.

### **CSCPPP (ponteiro)**

Este é o endereço da senha a ser usada na autenticação

Este é um campo de entrada. O valor inicial deste campo é o indicador nulo.

### **CSCSPUIL (número inteiro assinado de 10 dígitos)**

Este é o comprimento do ID do usuário a ser usado na autenticação

O comprimento máximo do ID do usuário não é dependente da plataforma Se o comprimento do ID do usuário for maior que o permitido, a solicitação de autenticação falhará com um RC2035.

Este é um campo de entrada. O valor inicial desse campo é 0.

### **CSCSPUIO (número inteiro assinado de 10 dígitos).**

Este é o deslocamento em bytes do ID do usuário a ser usado na autenticação.

O deslocamento pode ser positivo ou negativo

Este é um campo de entrada. O valor inicial desse campo é 0.

### **CSCSPUIP (ponteiro)**

Este é o endereço do ID do usuário a ser usado na autenticação

Este é um campo de entrada. O valor inicial deste campo é o indicador nulo. Esse campo será ignorado se o CSVER for menor que CSVER5

### **V 9.3.0 V 9.3.0 CSINITKL (número inteiro assinado de 10 dígitos)**

Este é o comprimento da chave inicial para o sistema de proteção de senha

Este é um campo de entrada. O valor inicial desse campo é 0.

### **V 9.3.0 V 9.3.0 CSINITKO (número inteiro assinado de 10 dígitos)**

Esse é o deslocamento em bytes da chave inicial para o sistema de proteção de senha. O deslocamento pode ser positivo ou negativo

É possível usar *CSINITKO* ou *CSINITKP* para especificar a chave inicial, mas não ambos. Para obter mais informações, consulte a descrição do campo *CSINITKP* ..

Este é um campo de entrada. O valor inicial desse campo é 0.

### **V 9.3.0 V 9.3.0 CSINITKP (ponteiro)**

Este é o endereço em bytes da chave inicial para o sistema de proteção de senha.

Este é um campo de entrada. O valor inicial deste campo é o indicador nulo.

IBM MQ MQI clients pode fornecer o valor de alguns campos como valores que foram criptografados usando o sistema de proteção de senha IBM MQ . Os campos a seguir podem conter valores criptografados:



- A senha do repositório de chaves na estrutura MQSCO.

Uma chave inicial é usada pelo algoritmo de criptografia para criptografar e descriptografar esses valores. Se uma chave inicial for fornecida quando os valores desses campos forem criptografados usando o utilitário **runmqicred**, então a mesma chave inicial deverá ser especificada pelo cliente quando ele se conectar ao gerenciador de fila

A chave inicial especificada usando esse campo substitui qualquer chave inicial especificada usando a variável de ambiente *MQS\_MQI\_KEYFILE* ou a propriedade *MQIInitialKeyFile* na sub-rotina Segurança do arquivo de configuração do cliente.

É possível usar *CSINITKO* ou *CSINITKP* para especificar a chave inicial, mas não ambos.

#### **CSRE1 (sequência de caracteres de 4 bytes)**

Um campo reservado, necessário para o alinhamento do ponteiro em IBM i

Este é um campo de entrada. O valor inicial deste campo é nulo.

#### **CSRS2 (sequência de caracteres de 8 bytes).**

Um campo reservado, necessário para o alinhamento do ponteiro em IBM i

Este é um campo de entrada. O valor inicial deste campo é nulo.

#### **CSSID (cadeia de caracteres de 4 bytes).**

Identificador de estruturação.

O valor deve ser:.

#### **CSSIDV**

Identificador para a estrutura de parâmetros de segurança.

#### **CSVER (número inteiro assinado de 10 dígitos)**

Número de versão da estrutura.

O valor deve ser:.

#### **CSVER1**

Estrutura de parâmetros de segurança Version-1 .



Estrutura de parâmetros de segurança Version-2 .

A constante a seguir especifica o número da versão atual:

#### **CSVERC**

A versão atual da estrutura de parâmetros de segurança







Isto é sempre um campo de entrada. O valor inicial desse campo é CSVER1.

### **Valores iniciais**

*Tabela 694. Campos em MQCNO*

<b>Nome de Campo</b>	<b>Nome da constante</b>	<b>Valor da constante</b>
<i>CSSID</i>	CSSIDV	' CSP↵ '
<i>CSVER</i>	CSVER1	1
<i>CSAUTHT</i>	Nenhum	0
<i>CSRE1</i>	Nenhum	Nulos
<i>CSCSPUIP</i>	Nenhum	Ponteiro nulo
<i>CSCSPUIO</i>	Nenhum	0

Tabela 694. Campos em MQCNO (continuação)

Nome de Campo	Nome da constante	Valor da constante
CSCSPUIIL	Nenhum	0
CSRS2	Nenhum	Nulos
CSCPPP	Nenhum	Ponteiro nulo
CSCPP0	Nenhum	0
CSCPPL	Nenhum	0
  CSINITKP	Nenhum	Ponteiro nulo
  CSINITKO	Nenhum	0
  CSINITKL	Nenhum	0

**Nota:**

1. O símbolo – representa um único caractere em branco.

**Declaração RPG**

```

D*..1.....2.....3.....4.....5.....6.....7..
D*
D* MQCSP Structure
D*
D* Structure identifier
D CSSID 1 4 INZ('CSP ')
D* Structure version number
D CSVER 5 8I 0 INZ(1)
D* Type of authentication
D CSAUTHT 9 12I 0 INZ(0)
D* Reserved
D CSRE1 13 16 INZ(X'00000000')
D* Address of user ID
D CSCSPUIP 17 32* INZ(*NULL)
D* Offset of user ID
D CSCSPUIO 33 36I 0 INZ(0)
D* Length of user ID
D CSCSPUIIL 37 40I 0 INZ(0)
D* Reserved
D CSRS2 41 48 INZ(X'0000000000000000')
D* Address of password
D CSCPPP 49 64* INZ(*NULL)
D* Offset of password
D CSCPP0 65 68I 0 INZ(0)
D* Length of password
D CSCPPL 69 72I 0 INZ(0)
    
```

 **MQCTLO (Estrutura de opções de retorno de chamada de controle) no IBM i**

Estrutura especificando a função de retorno de chamada de controle.

## Visão Geral

### Propósito

A estrutura MQCTLO é usada para especificar opções relacionadas a uma função de retorno de chamadas de controle.

A estrutura é um parâmetro de entrada e de saída na chamada [MQCTL](#)

### Versão

A versão atual do MQCTLO é CTLV1.

### Conjunto de caracteres e codificação

Os dados em MQCTLO devem estar no conjunto de caracteres fornecido pelo atributo do gerenciador de filas do **CodedCharSetId** e a codificação do gerenciador de fila local fornecido por ENNAT. No entanto, se o aplicativo estiver em execução como um cliente IBM MQ, a estrutura deverá estar no conjunto de caracteres e na codificação do cliente..

- “Campos” na página 1095
- “Valores iniciais” na página 1096
- “Declaração RPG” na página 1096

## Campos

A estrutura MQCTLO contém os seguintes campos; os campos são descritos em ordem alfabética:

### **COCONNAREA (número inteiro assinado de 10 dígitos)**

Estrutura de opções de controle-campo ConnectionArea

Este é um campo que está disponível para a função de retorno de chamada usar.

O gerenciador de filas não toma decisões com base no conteúdo desse campo e é transmitido inalterado a partir do campo [CBCCONNAREA](#) na estrutura MQCBC, que é um parâmetro na chamada MQCB.

Este campo é ignorado para todas as operações que não sejam CTLSR e CTLSW

Este é um campo de entrada e saída para a função de retorno de chamada. O valor inicial desse campo é um ponteiro nulo ou bytes nulos

### **COOPT (número inteiro assinado de 10 dígitos)**

Opções que controlam a ação de MQCTLO..

#### **CTLFQ**

Force a chamada MQCTLO a falhar se o gerenciador de filas ou a conexão estiver em quiesce.

Especifique GMFIQ, nas opções MQGMO transmitidas na chamada MQCB, para causar notificação aos consumidores de mensagens quando eles estiverem quiesce.

#### **CTLTHR**

Essa opção informa o sistema de que o aplicativo requer que todos os consumidores de mensagem, para a mesma conexão, sejam chamados no mesmo encadeamento

**Opção padrão:** se você não precisar de nenhuma das opções descritas, use a seguinte opção:

#### **CTLNO**

Use esse valor para indicar que nenhuma outra opção foi especificada. Todas as opções assumem seus valores-padrão. CTLNO é definido para auxiliar a documentação do programa; não se pretende que essa opção seja usada com qualquer outra, mas como seu valor é zero, tal uso não pode ser detectado.

Este é um campo de entrada. O valor inicial do campo *COOPT* é CTLNO.

### **CORSV (número inteiro assinado de 10 dígitos)**

Este é um campo reservado. O valor inicial desse campo é um caractere em branco.

### **COSID (número inteiro assinado de 10 dígitos)**

Estrutura de opções de controle-campo StrucId

Este é o identificador de estrutura; o valor deve ser:

#### **CTLSI**

Identificador para a estrutura de Opções de Controle

Isto é sempre um campo de entrada. O valor inicial desse campo é CTLSI.

### **COVER (número inteiro assinado de 10 dígitos)**

Estrutura de opções de controle-Campo de versão

Este é o número da versão da estrutura; o valor deve ser:

#### **CTLV1**

Version-1 Estrutura de opções de controle.

A constante a seguir especifica o número da versão atual:

#### **CTLCV**

A versão atual da estrutura de opções de Controle.

Isto é sempre um campo de entrada. O valor inicial desse campo é CTLV1.

## **Valores iniciais**

<i>Tabela 695. Campos em MQCTLO</i>		
<b>Nome de Campo</b>	<b>Nome da constante</b>	<b>Valor da constante</b>
<i>COSID</i>	CTLSI	'CTLO'
<i>COVER</i>	CTLV1	1
<i>COOPT</i>	CTLNO	Nulos
<i>CORSV</i>	Campo reservado	
<i>COCONNAREA</i>	Nenhum	Ponteiro nulo ou bytes nulos

## **Declaração RPG**

```
D* MQCTLO Structure
D*
D*
D* Structure identifier
D COSID          1      4   INZ('CTLO')
D*
D* Structure version number
D COVER          5      8I 0 INZ(1)
D*
D* Options that control the action of MQCTL
D COOPT          9      12I 0 INZ(0)
D*
D* Reserved
D CORSV          13     16I 0 INZ(-1)
D*
D* MQCTL Data area passed to the function
D COCONNAREA    17     32*  INZ(*NULL)
```

## **IBM i MQDH (Cabeçalho de distribuição) no IBM i**

A estrutura MQDH descreve os dados adicionais que estão presentes em uma mensagem quando essa mensagem é uma mensagem da lista de distribuição armazenada em uma fila de transmissão

## Visão Geral

**Propósito:** uma mensagem da lista de distribuição é uma mensagem que é enviada para várias filas de destino.. Os dados adicionais consistem na estrutura MQDH seguida por uma matriz de registros MQOR e uma matriz de registros MQPMR..

Essa estrutura é para uso por aplicativos especializados que colocam mensagens diretamente em filas de transmissão, ou que removem mensagens de filas de transmissão (por exemplo: agentes do canal de mensagem)

Essa estrutura não deve ser usada por aplicativos normais que simplesmente desejam colocar mensagens em listas de distribuições. Esses aplicativos devem usar a estrutura MQOD para definir os destinos na lista de distribuição e a estrutura MQPMO para especificar propriedades de mensagem ou receber informações sobre as mensagens enviadas para os destinos individuais..

**Conjunto de caracteres e codificação:** os dados em MQDH devem estar no conjunto de caracteres fornecido pelo atributo do gerenciador de filas **CodedCharSetId** e a codificação do gerenciador de fila local fornecido por ENNAT para a linguagem de programação C.

O conjunto de caracteres e a codificação do MQDH devem ser configurados nos campos *MDCSI* e *MDENC* em:

- O MQMD (se a estrutura MQDH estiver no início dos dados da mensagem) ou
- A estrutura do cabeçalho que precede a estrutura MQDH (todos os outros casos).

**Uso:** Quando um aplicativo coloca uma mensagem em uma lista de distribuições e alguns ou todos os destinos são remotos, o gerenciador de filas prefixa os dados da mensagem do aplicativo com as estruturas MQXQH e MQDH e coloca a mensagem na fila de transmissão relevante. Portanto, os dados ocorrem na sequência a seguir quando a mensagem está em uma fila de transmissão:

- estrutura MQXQH
- Estrutura MQDH mais matrizes de registros MQOR e MQPMR
- Dados da mensagem do aplicativo

Dependendo dos destinos, mais de uma mensagem desse tipo pode ser gerada pelo gerenciador de filas e colocada em filas de transmissão diferentes. Nesse caso, as estruturas MQDH nessas mensagens identificam subconjuntos diferentes dos destinos definidos pela lista de distribuição aberta pelo aplicativo

Um aplicativo que coloca uma mensagem da lista de distribuição diretamente em uma fila de transmissão deve estar em conformidade com a sequência descrita anteriormente e deve assegurar que a estrutura MQDH esteja correta.. Se a estrutura MQDH não for válida, o gerenciador de fila poderá optar por falhar a chamada MQPUT ou MQPUT1 com o código de razão RC2135.

As mensagens podem ser armazenadas em uma fila no formato de lista de distribuição somente se a fila for definida como capaz de suportar mensagens da lista de distribuição (consulte o atributo da fila **DistLists** descrito em [“Atributos para filas” na página 1416](#) ). Se um aplicativo colocar uma mensagem da lista de distribuição diretamente em uma fila que não suporta listas de distribuições, o gerenciador de filas dividirá a mensagem da lista de distribuição em mensagens individuais e as colocará na fila.

- [“Campos” na página 1097](#)
- [“Valores iniciais” na página 1100](#)
- [“Declaração RPG” na página 1101](#)

## Campos

A estrutura MQDH contém os campos a seguir; os campos são descritos em **ordem alfabética**:

### **DHCNT (número inteiro assinado de 10 dígitos)**

Número de registros do MQOR presentes

Isso define o número de destinos. Uma lista de distribuições deve sempre conter pelo menos um destino, portanto, *DHCNT* deve sempre ser maior que zero

O valor inicial desse campo é 0.

### **DHCSI (número inteiro assinado de 10 dígitos)**

Identificador do conjunto de caracteres de dados que segue os registros MQOR e MQPMR.

Isso especifica o identificador do conjunto de caracteres dos dados que seguem as matrizes de registros MQOR e MQPMR; ele não se aplica aos dados de caracteres na própria estrutura MQDH.

Na chamada MQPUT ou MQPUT1 , o aplicativo deve configurar esse campo para o valor apropriado aos dados. O seguinte valor especial pode ser usado:

#### **CSINHT**

Herde o identificador do conjunto de caracteres desta estrutura..

Os dados de caractere nos dados *após* essa estrutura estão no mesmo conjunto de caracteres que essa estrutura

O gerenciador de filas altera esse valor na estrutura enviada na mensagem para o identificador do conjunto de caractere real da estrutura.. Se nenhum erro ocorrer, o valor CSINHT não será retornado pela chamada MQGET.

CSINHT não poderá ser usado se o valor do campo *MDPAT* em MQMD for ATBRKR.

O valor inicial desse campo é CSUNDF.

### **DHENC (número inteiro assinado de 10 dígitos)**

A codificação numérica de dados que seguem os registros MQOR e MQPMR

Isso especifica a codificação numérica dos dados que seguem as matrizes de registros MQOR e MQPMR; ela não se aplica a dados numéricos na própria estrutura MQDH

Na chamada MQPUT ou MQPUT1 , o aplicativo deve configurar esse campo para o valor apropriado aos dados.

O valor inicial desse campo é 0.

### **DHFLG (número inteiro assinado de 10 dígitos)**

Sinalizadores gerais

A seguinte sinalização pode ser especificada:

#### **DHFNEW**

Gere novos identificadores de mensagem

Esse sinalizador indica que um novo identificador de mensagem deve ser gerado para cada destino na lista de distribuição. Isso pode ser configurado apenas quando não há registros put-message presentes ou quando os registros estão presentes, mas não contêm o campo *PRMID* .

Usar esse sinalizador adia a geração dos identificadores de mensagens até o último momento possível, ou seja, o momento em que a mensagem da lista de distribuição é finalmente dividida em mensagens individuais. Isso minimiza a quantidade de informações de controle que devem fluir com as mensagens da lista de distribuições.

Quando um aplicativo coloca uma mensagem em uma lista de distribuição, o gerenciador de filas configura DHFNEW no MQDH que ele gera quando ambas as instruções a seguir são verdadeiras:

- Não há nenhum registro put-message fornecido pelo aplicativo ou os registros fornecidos não contêm o campo *PRMID*
- O campo *MDMID* em MQMD é MINONE ou o campo *PMOPT* em MQPMO inclui PMNMID

Se nenhuma sinalização for necessária, o seguinte poderá ser especificado:

#### **DHFNON**

Sem bandeiras.

Esta constante indica que nenhuma sinalização foi especificada. DHFNON é definido para a documentação do programa de auxílio. Não se pretende que essa constante seja usada com nenhuma outra, mas como seu valor é zero, tal uso não pode ser detectado.

O valor inicial deste campo é DHFNON.

#### **DHFMT (cadeia de caracteres de 8 bytes).**

Nome do formato de dados que seguem os registros MQOR e MQPMR

Isso especifica o nome do formato dos dados que seguem as matrizes de registros MQOD e MQPMR (o que ocorrer por último).

Na chamada MQPUT ou MQPUT1, o aplicativo deve configurar esse campo para o valor apropriado aos dados. As regras para codificação desse campo são as mesmas que aquelas para o campo *MDFMT* no MQMD.

O valor inicial deste campo é FMNONE.

#### **DHLEN (número inteiro assinado de 10 dígitos)**

Comprimento da estrutura MQDH mais os registros MQOR e MQPMR.

Este é o número de bytes desde o início da estrutura MQDH até o início dos dados da mensagem após as matrizes de registros MQOR e MQPMR. Os dados ocorrem na seguinte sequência:

- Estrutura MQDH
- Matriz de registros MQOR
- Matriz de registros MQPMR
- Dados da mensagem

As matrizes de registros MQOR e MQPMR são tratadas por deslocamentos contidos na estrutura MQDH. Se esses deslocamentos resultarem em bytes não utilizados entre uma ou mais da estrutura MQDH, as matrizes de registros e os dados da mensagem, esses bytes não utilizados deverão ser incluídos no valor de *DHLEN*, mas o conteúdo desses bytes não será preservado pelo gerenciador de filas.. É válido para a matriz de registros MQPMR preceder a matriz de registros MQOR.

O valor inicial desse campo é 0.

#### **DHORO (número inteiro assinado de 10 dígitos)**

Deslocamento do primeiro registro MQOR do início de MQDH.

Esse campo fornece o deslocamento em bytes do primeiro registro na matriz de registros de objeto MQOR que contém os nomes das filas de destino. Há registros *DHCNT* nesta matriz. Esses registros (além de quaisquer bytes ignorados entre o primeiro registro de objeto e o campo anterior) são incluídos no comprimento fornecido pelo campo *DHLEN*.

Uma lista de distribuições deve sempre conter pelo menos um destino, portanto, *DHORO* deve sempre ser maior que zero.

O valor inicial desse campo é 0.

#### **DHPRF (número inteiro assinado de 10 dígitos)**

Sinalizadores indicando quais campos MQPMR estão presentes.

Zero ou mais dos seguintes sinalizadores podem ser especificados:

##### **PFMID**

O campo identificador de mensagem está presente.

##### **PFCID**

O campo do identificador de correlação está presente.

##### **PFGID**

O campo identificador de grupo está presente.

**PFFB**

Campo de feedback está presente.

**PFACC**

O campo token de contabilidade está presente.

Se nenhum campo MQPMR estiver presente, o seguinte poderá ser especificado:

**PFNONE**

Nenhum campo de registro de mensagem de colocação está presente

PFNONE é definido para a documentação do programa de auxílio Não se pretende que esta constante seja usada com qualquer outra, mas como seu valor é zero, tal uso não pode ser detectado.

O valor inicial deste campo é PFNONE.

**DHPRO (número inteiro assinado de 10 dígitos)**

Deslocamento do primeiro registro MQPMR do início de MQDH.

Esse campo fornece o deslocamento em bytes do primeiro registro na matriz de registros de mensagens put MQPMR contendo as propriedades de mensagem. Se estiver presente, há registros *DHCNT* nesta matriz Esses registros (mais quaisquer bytes ignorados entre o primeiro registro de mensagem de entrada e o campo anterior) são incluídos no comprimento fornecido pelo campo *DHLEN*

Registros de mensagens put são opcionais; se nenhum registro for fornecido, *DHPRO* será zero e *DHPRF* terá o valor PFNONE.

O valor inicial desse campo é 0.

**DHSID (cadeia de caracteres de 4 bytes)**

Identificador de estruturação.

O valor deve ser:.

**DHSIDV**

Identificador para a estrutura do cabeçalho de distribuição

O valor inicial deste campo é DHSIDV.

**DHVER (número inteiro assinado de 10 dígitos)**

Número de versão da estrutura.

O valor deve ser:.

**DHVER1**

Número da versão para estrutura do cabeçalho de distribuição.

A constante a seguir especifica o número da versão atual:

**DHVERC**

Versão atual da estrutura do cabeçalho de distribuição..

O valor inicial desse campo é DHVER1.

**Valores iniciais**

<i>Tabela 696. Campos em MQDH</i>		
<b>Nome de Campo</b>	<b>Nome da constante</b>	<b>Valor da constante</b>
<i>DHSID</i>	DHSIDV	'DH??'
<i>DHVER</i>	DHVER1	1
<i>DHLEN</i>	Nenhum	0



Tabela 696. Campos em MQDH (continuação)

Nome de Campo	Nome da constante	Valor da constante
DHENC	Nenhum	0
DHCSI	CSUNDF	0
DHFMT	FMNONE	Espaços em branco
DHFLG	DHFNON	0
DHPRF	PFNONE	0
DHCNT	Nenhum	0
DHORO	Nenhum	0
DHPRO	Nenhum	0

**Notas:**

1. O símbolo - representa um único caractere em branco.

**Declaração RPG**

```

D*..1.....2.....3.....4.....5.....6.....7..
D* MQDH Structure
D*
D* Structure identifier
D DHSID          1      4  INZ('DH ')
D* Structure version number
D DHVER          5      8I 0 INZ(1)
D* Length of MQDH structure plus following MQOR and MQPMP records
D DHLEN          9      12I 0 INZ(0)
D* Numeric encoding of data that follows the MQOR and MQPMP records
D DHENC         13     16I 0 INZ(0)
D* Character set identifier of data that follows the MQOR and MQPMP
D* records
D DHCSI         17     20I 0 INZ(0)
D* Format name of data that follows the MQOR and MQPMP records
D DHFMT         21     28  INZ(' ')
D* General flags
D DHFLG         29     32I 0 INZ(0)
D* Flags indicating which MQPMP fields are present
D DHPRF         33     36I 0 INZ(0)
D* Number of MQOR records present
D DHCNT         37     40I 0 INZ(0)
D* Offset of first MQOR record from start of MQDH
D DHORO         41     44I 0 INZ(0)
D* Offset of first MQPMP record from start of MQDH
D DHPRO         45     48I 0 INZ(0)
    
```

**IBM i MQDLH (cabeçalho de devoluções) em IBM i**

**Visão Geral**

**Propósito**

A estrutura MQDLH descreve as informações que prefixam os dados da mensagem do aplicativo de mensagens na fila de mensagens não entregues. Uma mensagem pode chegar na fila de mensagens não entregues porque o gerenciador de fila ou o agente do canal de mensagens a redirecionou para a fila... Um aplicativo pode colocar a mensagem diretamente em uma fila.

**Nome do Formato**

FMDLH

## Conjunto de caracteres e codificação

O MQDLH pode estar no início dos dados da mensagem do aplicativo. Nesse caso, os campos na estrutura MQDLH estão no conjunto de caracteres e na codificação fornecidos pelos campos MDCSI e MDENC. Se não, o conjunto de caracteres e a codificação serão configurados pelos campos MDCSI e MDENC na estrutura do cabeçalho que precede o MQDLH.

O conjunto de caracteres deve ser um que tenha caracteres de byte único para os caracteres válidos em nomes de filas.

## Uso

Os aplicativos que colocam mensagens diretamente na fila de mensagens não entregues devem prefixar os dados da mensagem com uma estrutura MQDLH e inicializar os campos com valores apropriados. No entanto, o gerenciador de fila não requer que uma estrutura MQDLH esteja presente ou que valores válidos sejam especificados para os campos.

Se uma mensagem for muito longa para ser colocada na fila de mensagens não entregues, o aplicativo deverá considerar executar uma das seguintes ações:

- Truncar os dados da mensagem para caber na fila de mensagens não entregues.
- Registre a mensagem no armazenamento auxiliar e coloque uma mensagem de relatório de exceções na fila de mensagens não entregues indicando que a mensagem é muito longa.
- Descartar a mensagem e retornar um erro para seu originador. Se a mensagem for uma mensagem crítica, descarte a mensagem somente se for conhecido que o originador ainda possui uma cópia da mensagem. Por exemplo, uma mensagem recebida por um agente do canal de mensagens de um canal de comunicação.

Qual das opções é apropriado depende do design da aplicação.

O gerenciador de filas executa processamento especial quando uma mensagem que é um segmento é colocada com uma estrutura MQDLH na frente. Consulte a descrição da estrutura do MQMDE para obter detalhes adicionais.

- [“Colocando mensagens na fila de mensagens não entregues” na página 1102](#)
- [“Obtendo mensagens da fila de mensagens não entregues” na página 1103](#)
- [“Campos” na página 1103](#)
- [“Valores iniciais” na página 1107](#)
- [“Declaração RPG” na página 1107](#)

## Colocando mensagens na fila de mensagens não entregues

Se uma mensagem for colocada na fila de mensagens não entregues, a estrutura MQMD usada para a chamada MQPUT ou MQPUT1 deverá ser idêntica à MQMD associada à mensagem. O MQMD geralmente é aquele retornado pela chamada MQGET, exceto para os casos a seguir:

- Os campos MDCSI e MDENC devem ser configurados para qualquer conjunto de caracteres e codificação são usados para campos na estrutura MQDLH.
- O campo MDFMT deve ser configurado para FMDLH para indicar que os dados começam com uma estrutura MQDLH.
- Os campos de contexto, MDACC, MDAID, MDAOD, MDPAN, MDPAT, MDPD, MDPTe MDUID devem ser configurados usando uma opção de contexto apropriada para as circunstâncias:
  - Um aplicativo que coloca na fila de mensagens não entregues uma mensagem não relacionada a nenhuma mensagem anterior deve usar a opção PMDEFC. A opção PMDEFC faz o gerenciador de filas configurar todos os campos de contexto no descritor de mensagens para seus valores padrão.
  - Um aplicativo do servidor que coloca na fila de mensagens não entregues uma mensagem recebida deve utilizar a opção PMPASA para preservar as informações de contexto originais.
  - Um aplicativo do servidor que coloca na fila de mensagens não entregues uma resposta à mensagem recebida deve usar a opção PMPASI. A opção PMPASI preserva as informações de identidade, mas configura as informações de origem para serem as do aplicativo do servidor.

- Um agente do canal de mensagens que coloca na fila de mensagens não entregues uma mensagem recebida de seu canal de comunicação deve usar a opção PMSETA . A opção PMSETA preserva as informações de contexto originais

Na própria estrutura MQDLH , os campos devem ser configurados da seguinte forma:

- Os campos DLCSI, DLENCE *DLFMT* devem ser configurados para os valores que descrevem os dados que seguem a estrutura MQDLH Esses valores geralmente são os valores do descritor de mensagem original
- Os campos de contexto DLPAT, DLPAN, DLPDe DLPT devem ser configurados para valores apropriados para o aplicativo que está colocando a mensagem na fila de mensagens não entregues. Esses valores não estão relacionados com a mensagem original
- Outros campos devem ser configurados conforme apropriado..

O aplicativo deve assegurar que todos os campos tenham valores válidos e que os campos de caracteres sejam preenchidos com espaços em branco para o comprimento definido do campo. Os dados de caractere não devem ser encerrados prematuramente usando um caractere nulo.. O gerenciador de filas não converte os caracteres nulos e subsequentes em espaços em branco na estrutura MQDLH .

## Obtendo mensagens da fila de mensagens não entregues

Aplicativos que obtêm mensagens da fila de mensagens não entregues devem verificar se as mensagens começam com uma estrutura MQDLH . O aplicativo pode determinar se uma estrutura MQDLH está presente, examinando o campo MDF MT no descritor de mensagem MQMD Se o campo tiver o valor FMDLH, os dados da mensagem começam com uma estrutura MQDLH . As mensagens na fila de mensagens não entregues podem ser truncadas se originalmente eram muito longas para a fila para a qual foram destinadas.

## Campos

A estrutura MQDLH contém os campos a seguir; os campos são descritos em ordem alfabética:

### **DLCSI (número inteiro assinado de 10 dígitos)**

Identificador do conjunto de caracteres de dados que seguem MQDLH.

DLCSI especifica o identificador do conjunto de dados que segue a estrutura MQDLH . Os dados geralmente são da mensagem original. Ele não se aplica aos dados de caracteres na própria estrutura do MQDLH

Na chamada MQPUT ou MQPUT1 , o aplicativo deve configurar esse campo para o valor apropriado aos dados. O seguinte valor especial pode ser usado:

#### **CSINHT**

Herde o identificador do conjunto de caracteres desta estrutura..

Os dados de caracteres nos dados que seguem esta estrutura estão no mesmo conjunto de caracteres que esta estrutura

O gerenciador de filas altera esse valor na estrutura enviada na mensagem para o identificador do conjunto de caractere real da estrutura.. Contanto que nenhum erro ocorra, o valor CSINHT não será retornado pela chamada MQGET .

CSINHT não poderá ser usado se o valor do campo MDPAT em MQMD for ATBRKR.

O valor inicial desse campo é CSUNDF.

### **DLDM (sequência de caracteres de 48 bytes)**

Nome do gerenciador de filas de origem.

Esse é o nome do gerenciador de filas que era o destino original da mensagem

O comprimento desse campo é fornecido por LNQMN O valor inicial desse campo é 48 caracteres em branco.

**DLDQ (cadeia de caracteres de 48 bytes).**

Nome da fila de destino original..

Este é o nome da fila de mensagens que era o destino original para a mensagem

O comprimento desse campo é fornecido por LNQN O valor inicial desse campo é 48 caracteres em branco.

**DLENC (número inteiro assinado de 10 dígitos)**

Codificação numérica de dados que seguem MQDLH..

DLENC especifica a codificação numérica dos dados após a estrutura MQDLH . Os dados geralmente são da mensagem original. Ele não se aplica a dados numéricos na própria estrutura do MQDLH

Na chamada MQPUT ou MQPUT1 , o aplicativo deve configurar esse campo para o valor apropriado aos dados.

O valor inicial desse campo é 0.

**DLFMT (cadeia de caracteres de 8 bytes).**

Formatar o nome dos dados que seguem MQDLH

Isso especifica o nome do formato dos dados que seguem a estrutura MQDLH (geralmente os dados da mensagem original).

Na chamada MQPUT ou MQPUT1 , o aplicativo deve configurar esse campo para o valor apropriado aos dados. As regras para codificar esse campo são as mesmas que as regras para o campo MDFMT no MQMD

O comprimento desse campo é fornecido por LNFMT O valor inicial desse campo é FMNONE.

**DLPAN (sequência de caracteres de 28 bytes)**

Nome do aplicativo que coloca a mensagem na fila de mensagens não entregues.

O formato do nome depende do campo DLPAT .. Consulte a descrição do campo MDPAN em [“MQMD \(Descritor de mensagens\) em IBM i” na página 1148](#)

Se for o gerenciador de filas que redireciona a mensagem para a fila de mensagens não entregues, DLPAN contém os primeiros 28 caracteres do nome do gerenciador de filas. O nome é preenchido com espaços em branco se necessário.

O comprimento desse campo é fornecido por LNPAN O valor inicial deste campo é de 28 caracteres em branco.

**DLPAT (número inteiro assinado de 10 dígitos)**

Tipo de aplicativo que coloca a mensagem na fila de mensagens não entregues (não entregues).

Esse campo tem o mesmo significado que o campo MDPAT no descritor de mensagem MQMD (consulte [“MQMD \(Descritor de mensagens\) em IBM i” na página 1148](#) para obter detalhes.

Se for o gerenciador de filas que redireciona a mensagem para a fila de mensagens não entregues, DLPAT terá o valor ATQM.

O valor inicial desse campo é 0.

**DLPD (cadeia de caracteres de 8 bytes)**

Data em que a mensagem foi colocada na fila de mensagens não entregues.

O formato utilizado para a data em que esse campo é gerado pelo gerenciador de filas é:

• YYYYMMDD

em que os caracteres representam:

**YYYY**

ano (quatro dígitos numéricos)

**MM**

mês do ano (01 a 12)

**DD**

dia do mês (01 a 31)

O Tempo Médio de Greenwich (GMT) é usado para os campos DLPD e DLPT , sujeito ao relógio do sistema sendo configurado com precisão para GMT.

O comprimento desse campo é fornecido por LNPDAT O valor inicial deste campo é oito caracteres em branco.

**DLPT (cadeia de caracteres de 8 bytes).**

Hora em que a mensagem foi colocada na fila de mensagens não entregues.

O formato usado para o horário em que esse campo é gerado pelo gerenciador de filas é:

- HHMMSSSTH

em que os caracteres representam (por ordem):

**HH**

horas (00 a 23)

**MM**

minutos (00 a 59)

**SS**

segundos (00 a 59; ver nota mais tarde neste tópico)

**T**

Décimos de segundo (0 a 9)

**H**

centésimos de segundo (0 a 9)

**Nota:** Se o relógio do sistema estiver sincronizado com um padrão de tempo exato, será possível que 60 ou 61 sejam retornados para os segundos em DLPT O segundo extra ocorre quando os segundos bissextos são inseridos no padrão de tempo global

O Tempo Médio de Greenwich (GMT) é usado para os campos DLPD e DLPT , sujeito ao relógio do sistema sendo configurado com precisão para GMT.

O comprimento desse campo é fornecido por LNPTIM O valor inicial deste campo é oito caracteres em branco.

**DLREA (número inteiro assinado de 10 dígitos)**

A mensagem de razão chegou na fila de mensagens não entregues.

Isso identifica o motivo pelo qual a mensagem foi colocada na fila de devoluções em vez de na fila de destino original. Deve ser um dos valores FB\* ou RC\* (por exemplo, RC2053). Consulte a descrição do campo *MDFB* em [“MQMD \(Descritor de mensagens\) em IBM i”](#) na página 1148 para obter detalhes dos valores comuns de FB\* que podem ocorrer

Se o valor estiver no intervalo de FBIFST a FBILST, o código de erro IMS real poderá ser determinado subtraindo FBIERR do valor do campo *DLREA* .

Alguns valores de FB\* ocorrem apenas neste campo Eles se relacionam a mensagens do repositório, mensagens do acionador ou mensagens da fila de transmissão que são transferidas para a fila de mensagens não entregues Esses valores são:

**FBABEG**

O aplicativo não pode ser iniciado..

Um aplicativo processando uma mensagem do acionador não pôde iniciar o aplicativo nomeado no campo TMAI da mensagem do acionador; consulte [“MQTM-Mensagem do acionador”](#) na página 1280.

**FBATYP**

Erro de tipo de aplicativo

Um aplicativo processando uma mensagem do acionador não pôde iniciar o aplicativo porque o campo TMTAT da mensagem do acionador não é válido; consulte [“MQTM-Mensagem do acionador” na página 1280.](#)

**FBB OCD**

Canal do receptor de clusters excluído.

A mensagem estava em uma fila de transmissão do cluster destinada a uma fila de clusters que foi aberta com a opção FBIERR. O canal do receptor de clusters remoto a ser usado para transmitir a mensagem para a fila de destino foi excluído antes que a mensagem pudesse ser enviada. Como FBIERR foi especificado, apenas o canal selecionado quando a fila foi aberta pode ser usado para transmitir a mensagem... Como esse canal não está mais disponível, a mensagem foi colocada na fila de mensagens não entregues.

**FBNARM**

A mensagem não é uma mensagem de repositório

**FBSBCX**

Mensagem interrompida pela saída de definição automática do canal.

**FBSBMX**

Mensagem interrompida pela saída de mensagem do canal.

**FBTM**

Estrutura MQTM inválida ou ausente.

O campo MDFMT em MQMD especifica FMTM, mas a mensagem não começa com uma estrutura MQTM válida. Por exemplo, o destaque mnemônico *TMSID* pode não ser válido. O *TMVER* pode não ser reconhecido.. O comprimento da mensagem acionador pode ser insuficiente para conter a estrutura MQTM.

**FBXQME**

Mensagem na fila de transmissão não está no formato correto.

Um agente do canal de mensagens descobriu que uma mensagem na fila de transmissão não está no formato correto. O agente do canal de mensagens coloca a mensagem na fila de mensagens não entregues usando esse código de feedback...

O valor inicial desse campo é RCNONE.

**DLSID (sequência de caracteres de 4 bytes)**

Identificador de estruturação.

O valor deve ser:.

**DLSIDV**

Identificador para a estrutura do cabeçalho de devoluções.

O valor inicial desse campo é DLSIDV.

**DLVER (número inteiro assinado de 10 dígitos).**

Número de versão da estrutura.

O valor deve ser:.

**DLVER1**

Número da versão para a estrutura do cabeçalho de devoluções

A constante a seguir especifica o número da versão atual:

**DLVERC**

A versão atual da estrutura do cabeçalho de devoluções

O valor inicial desse campo é DLVER1.

## Valores iniciais

Tabela 697. Campos em MQDLH		
Nome de Campo	Nome da constante	Valor da constante
DLSID	DLSIDV	'DLH~'
DLVER	DLVER1	1
DLREA	RCNONE	0
DLDQ	Nenhum	Espaços em branco
DLDM	Nenhum	Espaços em branco
DLENC	Nenhum	0
DLCSI	CSUNDF	0
DLFMT	FMNONE	Espaços em branco
DLPAT	Nenhum	0
DLPAN	Nenhum	Espaços em branco
DLPD	Nenhum	Espaços em branco
DLPT	Nenhum	Espaços em branco

### Notas:

1. O símbolo ~ representa um único caractere em branco.

## Declaração RPG

```

D*.1.....2.....3.....4.....5.....6.....7..
D* MQDLH Structure
D*
D* Structure identifier
D DLSID          1      4    INZ('DLH ')
D* Structure version number
D DLVER          5      8I 0 INZ(1)
D* Reason message arrived on dead-letter(undelivered-message) queue
D DLREA          9     12I 0 INZ(0)
D* Name of original destination queue
D DLDQ           13     60    INZ
D* Name of original destination queue manager
D DLDM           61     108   INZ
D* Numeric encoding of data that followsMQDLH
D DLENC          109    112I 0 INZ(0)
D* Character set identifier of data thatfollows MQDLH
D DLCSI          113    116I 0 INZ(0)
D* Format name of data that followsMQDLH
D DLFMT          117    124   INZ(' ')
D* Type of application that put messageon dead-letter
D* (undelivered-message)queue
D DLPAT          125    128I 0 INZ(0)
D* Name of application that put messageon dead-letter
D* (undelivered-message)queue
D DLPAN          129    156   INZ
D* Date when message was put ondead-letter (undelivered-message)queue
D DLPD           157    164   INZ
D* Time when message was put on thedead-letter (undelivered-message)queue
D DLPT           165    172   INZ

```

## MQDMHO (Excluir opções de manipulação de mensagem) em IBM i

A estrutura **MQDMHO** permite que os aplicativos especifiquem opções que controlam como as manipulações de mensagens são excluídas

## Visão Geral

**Propósito:** A estrutura é um parâmetro de entrada na chamada **MQDLTMH**

**Conjunto de caracteres e codificação:** os dados em **MQDMHO** devem estar no conjunto de caracteres do aplicativo e na codificação do aplicativo (ENNAT).

- “Campos” na página 1108
- “Valores iniciais” na página 1108
- “Declaração RPG” na página 1108

## Campos

A estrutura MQDMHO contém os seguintes campos; os campos são descritos em **ordem alfabética**:

### **DMOPT (número inteiro assinado de 10 dígitos)**

O valor deve ser:.

#### **DMNONE**

Nenhuma opção especificada.

Isto é sempre um campo de entrada. O valor inicial desse campo é **DMNONE**.

### **DMSID (número inteiro assinado de 10 dígitos)**

Este é o identificador de estrutura; o valor deve ser:

#### **DMSIDV**

Identificador para excluir estrutura de opções de manipulação de mensagens.

Isto é sempre um campo de entrada. O valor inicial desse campo é **DMSIDV**.

### **DMVER (número inteiro assinado de 10 dígitos)**

Este é o número da versão da estrutura; o valor deve ser:

#### **DMVER1**

Version-1 exclua a estrutura de opções de manipulação de mensagens

A constante a seguir especifica o número da versão atual:

#### **DMVERC**

A versão atual da estrutura de opções de manipulação de mensagens de exclusão

Isto é sempre um campo de entrada. O valor inicial desse campo é **DMVER1**.

## Valores iniciais

<b>Nome de Campo</b>	<b>Nome da constante</b>	<b>Valor da constante</b>
<i>DMSID</i>	DMSIDV	' DMHO '
<i>DMVER</i>	DMVER1	1
<i>DMOPT</i>	DMNONE	0

## Declaração RPG

D\* MQDMHO Structure



```

D*
D*
D* Structure identifier
D DMSID          1      4    INZ('DMHO')
D*
D* Structure version number
D DMVER          5      8I 0 INZ(1)
D*
D* Options that control the action of MQDLTMH
D DMOPT          9      12I 0 INZ(0)

```

## IBM i MQDMPO (Excluir opções de propriedade de mensagem) em IBM i

Estrutura que define as opções de propriedade de mensagem de exclusão

### Visão Geral

**Propósito:** A estrutura MQDMPO permite que aplicativos especifiquem opções que controlam como as propriedades de mensagens são excluídas. A estrutura é um parâmetro de entrada na chamada de MQDLTMP

**Conjunto de caracteres e codificação:** os dados no MQDMPO devem estar no conjunto de caracteres do aplicativo e na codificação do aplicativo (ENNAT)..

- [“Campos” na página 1109](#)
- [“Valores iniciais” na página 1110](#)
- [“Declaração RPG” na página 1110](#)

### Campos

A estrutura MQDMPO contém os campos a seguir; os campos são descritos em ordem alfabética:

#### DPOPT (número inteiro assinado de 10 dígitos)

Excluir estrutura de opções de propriedade de mensagem-campo DPOPT.

**Opções de local:** As opções a seguir estão relacionadas ao local relativo da propriedade comparado ao cursor da propriedade.

#### DPDEL

Exclui a primeira propriedade que corresponde ao nome especificado

#### DPDEL

Exclui a propriedade apontada pelo cursor de propriedade; essa é a propriedade que foi consultada por último usando a opção IPINQF ou IPINQN.

O cursor de propriedade é reconfigurado quando o identificador de mensagem é reutilizado Ele também é reconfigurado quando a manipulação de mensagens é especificada no campo *HMSG* do MQGMO em uma chamada MQGET ou na estrutura MQPMO em uma chamada MQPUT

O cursor de propriedade é reconfigurado quando a manipulação da mensagem é reutilizada, ou quando a manipulação da mensagem é especificada no campo *HMSG* da estrutura MQGMO em uma estrutura MQGET em uma chamada MQGET ou na estrutura MQPMO em uma chamada MQPUT

A chamada falha com o código de conclusão CCFAIL e motivo RC2471 se essa opção for usada quando o cursor da propriedade ainda não tiver sido estabelecido. Também falha com esses códigos se a propriedade apontada pelo cursor da propriedade já tiver sido excluída.

Se nenhuma dessas opções for necessária, a seguinte opção poderá ser usada:

#### DPNONE

Nenhuma opção especificada.

O valor inicial deste campo de entrada é DPDEL.

### DPSID (número inteiro assinado de 10 dígitos)

Excluir a estrutura de opções de propriedades de mensagem-campo DPSID.

Este é o identificador de estrutura. O valor deve ser:.

#### DPSIDV

Identificador para excluir a estrutura de opções de propriedade de mensagem.

Esse campo é sempre um campo de entrada O valor inicial desse campo é DPSIDV.

### DPVER (número inteiro assinado de 10 dígitos).

Excluir estrutura de opções de propriedade de mensagem-campo DPVER.

Este é o número da versão da estrutura O valor deve ser:.

#### DPVER1

Número da versão para a estrutura de opções de propriedades de mensagens de exclusão

A constante a seguir especifica o número da versão atual:

#### DPVERC

Versão atual da estrutura de opções de propriedades de mensagem de exclusão

Esse campo é sempre um campo de entrada O valor inicial desse campo é DPVER1.

## Valores iniciais

Tabela 699. Campos em MQDPMO		
Nome de Campo	Nome da constante	Valor da constante
DPSID	DPSIDV	'DMPO'
DPVER	DPVER1	1
DPOPT	Opções que controlam a ação de MQDLTMP	DPNONE

## Declaração RPG

```
D* MQDMPO Structure
D*
D*
D* Structure identifier
D  DPSID          1      4  INZ('DMPO')
D*
D* Structure version number
D  DPVER          5      8I 0 INZ(1)
D*
** Options that control the action of
D* MQDLTMP
D  DPOPT          9     12I 0 INZ(0)
```

## MQEPH (cabeçalho PCF integrado) no IBM i

### Visão Geral

#### Propósito

A estrutura MQEPH descreve os dados adicionais presentes em uma mensagem quando essa mensagem é uma mensagem de formato de comando programável (PCF). O campo *EPPFH* define os parâmetros PCF que seguem essa estrutura e isso permite seguir os dados da mensagem PCF com outros cabeçalhos.

#### Nome do Formato

EPFMT

## Conjunto de caracteres e codificação

Os dados em MQEPH devem estar no conjunto de caracteres e na codificação do gerenciador de filas locais; isso é fornecido pelo atributo do gerenciador de filas do **CCSID**

Configure o conjunto de caracteres e a codificação do MQEPH nos campos *MDCSI* e *MDENC* em:

- O MQMD (se a estrutura MQEPH estiver no início dos dados da mensagem) ou
- A estrutura do cabeçalho que precede a estrutura MQEPH (todos os outros casos).

## Uso

Não é possível usar estruturas MQEPH para enviar comandos para o servidor de comando ou qualquer outro servidor de aceitação PCF do gerenciador de filas.

Da mesma forma, o servidor de comandos ou qualquer outro servidor de aceitação PCF do gerenciador de filas não gera respostas ou eventos contendo estruturas MQEPH.

- [“Campos” na página 1111](#)
- [“Valores iniciais” na página 1112](#)
- [“Declaração RPG” na página 1113](#)

## Campos

A estrutura MQEPH contém os seguintes campos; os campos são descritos em **ordem alfabética**:

### EPCSI (número inteiro assinado de 10 dígitos)

Esse é o identificador do conjunto de caracteres dos dados que seguem a estrutura MQEPH e os parâmetros PCF associados; ele não se aplica aos dados de caracteres na própria estrutura MQEPH

O valor inicial deste campo é EPCUND.

### EPENC (número inteiro assinado de 10 dígitos)

Esta é a codificação numérica dos dados que seguem a estrutura MQEPH e os parâmetros PCF associados; ela não se aplica aos dados de caracteres na própria estrutura MQEPH

O valor inicial desse campo é 0.

### EPFLG (número inteiro assinado de 10 dígitos)

Os seguintes valores estão disponíveis:

#### EPNONE

Nenhuma sinalização foi especificada. *MDCSI* O EPNONE é definido para auxiliar a documentação do programa Não se pretende que esta constante seja usada com qualquer outra, mas como seu valor é zero, tal uso não pode ser detectado.

#### EPCSEM

O conjunto de caracteres dos parâmetros que contêm dados de caracteres é especificado individualmente dentro do campo *CCSID* em cada estrutura O conjunto de caracteres dos campos *EPSID* e *EPFMT* é definido pelo *CCSID* na estrutura do cabeçalho que precede a estrutura MQEPH, ou pelo campo *MDCSI* no MQMD se o MQEPH estiver no início da mensagem

O valor inicial deste campo é EPNONE.

### EPFMT (cadeia de caracteres de 8 bytes)

Este é o nome do formato dos dados que seguem a estrutura MQEPH e os parâmetros PCF associados

O valor inicial deste campo é EPFMNO.

### EPLEN (número inteiro assinado de 10 dígitos)

Esta é a quantidade de dados que precedem a próxima estrutura do cabeçalho... Ele inclui:

- O comprimento do cabeçalho MQEPH
- O comprimento de todos os parâmetros PCF após o cabeçalho

- Qualquer preenchimento em branco seguindo esses parâmetros

EPLEN deve ser um múltiplo de 4.

A parte de comprimento fixo da estrutura é definida por EPSTLF

O valor inicial desse campo é 68.

### EPPCFH (MQCFH)

Este é o cabeçalho PCF (Programmable Command Format), definindo os parâmetros PCF que seguem a estrutura MQEPH. Isso permite seguir os dados da mensagem PCF com outros cabeçalhos.

O cabeçalho PCF é inicialmente definido com os seguintes valores:

<i>Tabela 700. Campos em EPPCFH</i>		
<b>Nome de Campo</b>	<b>Nome da constante</b>	<b>Valor da constante</b>
EP3TYP	CFTNON	0
EP3LEN	FHLENV	36
EP3VER	FHVER3	3
EP3CMD	CMNONE	0
EP3SEQ	Nenhum	1
EP3CTL	CFCLST	1
EEP3CC	CCOK	0
EP3REA	RCNONE	0
EP3CNT	Nenhum	0

O aplicativo deve mudar EP3TYP de CFTNON para um tipo de estrutura válido para o uso que está fazendo do cabeçalho PCF integrado.

### EPSID (sequência de caracteres de 4 bytes).

O valor deve ser:.

#### EPSTID

Identificador para a estrutura do cabeçalho PCF integrado.

O valor inicial desse campo é EPSTID.

### EPVER (número inteiro assinado de 10 dígitos)

O valor do pode ser:

#### EPVER1

Número da versão para a estrutura do cabeçalho PCF integrado

A constante a seguir especifica o número da versão atual:

#### EPVER3

A versão atual da estrutura do cabeçalho PCF integrado

O valor inicial desse campo é EPVER3.

### Valores iniciais

<i>Tabela 701. Campos em MQEPH</i>		
<b>Nome de Campo</b>	<b>Nome da constante</b>	<b>Valor da constante</b>
EPSID	EPSTID	'EPST'

Tabela 701. Campos em MQEPH (continuação)

Nome de Campo	Nome da constante	Valor da constante
EPVER	EPVER1	1
EPLEN	EPSTLF	68
EPENC	Nenhum	0
EPCSI	EPCUND	0
EPFMT	EPFMNO	Espaços em branco
EPFLG	EPNONE	0
EPPCFH	Nomes e valores conforme definido em Tabela 700 na página 1112	0

**Nota:**

1. O símbolo – representa um único caractere em branco.

**Declaração RPG**

```

D*..1.....2.....3.....4.....5.....6.....7..
D* MQEPH Structure
D*
D* Structure identifier
D  EPSID          1          4
D* Structure version number
D  EPVER          5          8I 0
D* Total length of MQEPH including MQCFHand parameter structures
D* that follow
D  EPLEN          9          12I 0
D* Numeric encoding of data that follows last PCF parameter structure
D  EPENC          13         16I 0
D* Character set identifier of data that follows last PCF parameter
D* structure
D  EPCSI          17         20I 0
D* Format name of data that follows last PCF parameter structure
D  EPFMT          21         28
D* Flags
D  EPFLG          29         32I 0
D* Programmable Command Format Header
D  EP3TYP          33         36I 0
D  EP3LEN          37         40I 0
D  EP3VER          41         44I 0
D  EP3CMD          45         48I 0
D  EP3SEQ          49         52I 0
D  EP3CTL          53         56I 0
D  EP3CC           57         60I 0
D  EP3REA          61         64I 0
D  EP3CNT          65         68I 0
    
```

**IBM i MQGMO (opções Get-message) em IBM i**

A estrutura MQGMO permite que o aplicativo especifique opções que controlam como as mensagens são removidas das filas..

**Visão Geral**

**Propósito**

A estrutura é um parâmetro de entrada / saída na chamada MQGET..

**Versão**

A versão atual do MQGMO é GMVER4 Os campos que existem apenas nas versões mais recentes da estrutura são identificados como tal nas descrições a seguir:

O arquivo COPY fornecido contém a versão mais recente de MQGMO suportada pelo ambiente, mas com o valor inicial do campo *GMVER* configurado como GMVER1. Para usar campos que não estão presentes na estrutura version-1 , o aplicativo deve configurar o campo *GMVER* para o número da versão necessária.

### **Conjunto de caracteres e codificação**

Os dados em MQGMO devem estar no conjunto de caracteres fornecido pelo atributo do gerenciador de filas do **CodedCharSetId** e pela codificação do gerenciador de fila local fornecido pelo ENNAT. No entanto, se o aplicativo estiver em execução como um cliente IBM MQ , a estrutura deverá estar no conjunto de caracteres e na codificação do cliente..

- [“Campos” na página 1114](#)
- [“Valores iniciais” na página 1135](#)
- [“Declaração RPG” na página 1135](#)

## **Campos**

A estrutura MQGMO contém os seguintes campos; os campos são descritos em ordem alfabética:

### **GMGST (cadeia de caracteres de 1 byte)**

Sinalizador que indica se a mensagem recuperada está em um grupo

Ele possui um dos seguintes valores:

#### **GSNIG**

A mensagem não está em um grupo.

#### **GSMIG**

A mensagem está em um grupo, mas não é a última no grupo.

#### **GSLMIG**

A mensagem é a última no grupo.

Esse valor também será o valor retornado se o grupo consistir em apenas uma mensagem..

Este campo é um campo de saída. O valor inicial deste campo é GSNIG. Esse campo será ignorado se *GMVER* for menor que GMVER2.

### **GMMH (número inteiro assinado de 10 dígitos)**

Identificador de mensagens

Se a opção GMPRAQ for especificada e o atributo da fila PRPCTL não for configurado como PRPRFH, este será o identificador para uma mensagem que é preenchida com as propriedades da mensagem que está sendo recuperada da fila. O identificador é criado por uma chamada MQCRTM. Todas as propriedades já associadas ao identificador são limpas antes de recuperar uma mensagem.

O seguinte valor também pode ser especificado:

MQHM\_NONE

Nenhuma manipulação de mensagem fornecida.

Nenhum descritor de mensagens é necessário na chamada MQGET se um identificador de mensagens válido for fornecido e usado na saída para conter as propriedades de mensagem, o descritor de mensagens associado ao identificador de mensagens é usado para campos de entrada

Se um descritor de mensagens for especificado na chamadas MQGET, ele sempre terá precedência sobre o descritor de mensagem associado a um identificador de mensagens

Se o GMPRRF for especificado ou o GMPRAQ for especificado e o atributo da fila PRPCTL for PRPRFH, a chamada falhará com o código de razão RC2026 quando nenhum parâmetro do descritor de mensagens for especificado.

No retorno da chamada MQGET, as propriedades e o descritor de mensagens associados a essa manipulação de mensagens são atualizados para refletir o estado da mensagem recuperada (bem

como o descritor de mensagens se um foi fornecido na chamada MQGET). As propriedades da mensagem podem então ser consultadas usando a chamada MQINQMP.

Exceto para as extensões do descritor de mensagens, quando presentes, uma propriedade que pode ser consultada com a chamada MQINQMP não está contida nos dados da mensagem; se a mensagem na fila continha propriedades nos dados da mensagem, eles serão removidos dos dados da mensagem antes que os dados sejam retornados ao aplicativo.

Se nenhuma manipulação de mensagem for fornecida, ou a versão for menor que GMVER4, você deverá fornecer um descritor de mensagens válido na chamada MQGET. Quaisquer propriedades de mensagem (exceto aquelas contidas no descritor de mensagens) são retornadas no assunto de dados de mensagem para o valor das opções de propriedade na estrutura MQGMO e no atributo da fila PRPCTL..

Esse campo é sempre um campo de entrada.. O valor inicial deste campo é HMNONE. Esse campo será ignorado se *GMVER* for menor que GMVER4

### **GMMO (número inteiro assinado de 10 dígitos)**

Opções que controlam os critérios de seleção usados para MQGET.

Essas opções permitem que o aplicativo escolha quais campos no parâmetro **MSGDSC** são usados para selecionar a mensagem retornada pela chamada MQGET. O aplicativo configura as opções necessárias nesse campo, em seguida, configura os campos correspondentes no parâmetro **MSGDSC** para os valores necessários para esses campos. Apenas as mensagens que possuem esses valores no MQMD para a mensagem são candidatas para recuperação usando esse parâmetro **MSGDSC** na chamada MQGET. Os campos para os quais a opção correspondente não é especificada são ignorados ao selecionar a mensagem a ser retornada. Se nenhum critério de seleção for usado na chamada MQGET (ou seja, qualquer mensagem é aceitável), *GMMO* deverá ser configurado como MONONE.

Se GMLOGO for especificado, apenas determinadas mensagens serão elegíveis para retorno pela próxima chamada MQGET:

- Se não houver nenhum grupo atual ou mensagem lógica, apenas as mensagens que tiverem *MDSEQ* igual a 1 e *MDOFF* igual a 0 serão elegíveis para retorno. Nessa situação, uma ou mais das opções a seguir podem ser usadas para selecionar qual das mensagens elegíveis é aquela retornada:
  - MOMSGI
  - MOCORI
  - MOGRPI
- Se houver um grupo atual ou mensagem lógica, apenas a próxima mensagem no grupo ou próximo segmento na mensagem lógica será elegível para retorno e isso não poderá ser alterado especificando as opções MO\*.

Em ambos os casos, as opções de correspondência que não são aplicáveis ainda podem ser especificadas, mas o valor do campo relevante no parâmetro **MSGDSC** deve corresponder ao valor do campo correspondente na mensagem a ser retornada; a chamada falha com o código de razão RC2247 se essa condição não for atendida.

*GMMO* será ignorado se GMMUC ou GMBRWC for especificado..

Uma ou mais das seguintes opções podem ser especificadas:

#### **MOMSGI**

Recuperar mensagem com o identificador de mensagem especificado.

Esta opção especifica que a mensagem a ser recuperada deve ter um identificador de mensagens que corresponda ao valor do campo *MDMID* no parâmetro **MSGDSC** da chamada MQGET.. Essa correspondência é adicional a quaisquer outras correspondências que possam se aplicar (por exemplo, o identificador de correlação).

Se esta opção não for especificada, o campo *MDMID* no parâmetro **MSGDSC** será ignorado e qualquer identificador de mensagem corresponderá.

**Nota:** O identificador de mensagem MINONE é um valor especial que corresponde a qualquer identificador de mensagem no MQMD da mensagem. Portanto, especificar MOMSGI com MINONE é o mesmo que não especificar MOMSGI.

### **MOCORI**

Recuperar mensagem com o identificador de correlação especificado.

Esta opção especifica que a mensagem a ser recuperada deve ter um identificador de correlação que corresponde ao valor do campo *MDCID* no parâmetro **MSGDSC** da chamada MQGET. Essa correspondência é adicional a quaisquer outras correspondências que possam se aplicar (por exemplo, o identificador de mensagem).

Se essa opção não for especificada, o campo *MDCID* no parâmetro **MSGDSC** será ignorado e qualquer identificador de correlação será correspondente.

**Nota:** O identificador de correlação CINONE é um valor especial que corresponde a qualquer identificador de correlação no MQMD para a mensagem. Portanto, especificar MOCORI com CINONE é o mesmo que não especificar MOCORI.

### **MOGRPI**

Recuperar mensagem com o identificador de grupo especificado.

Esta opção especifica que a mensagem a ser recuperada deve ter um identificador de grupo que corresponda ao valor do campo *MDGID* no parâmetro **MSGDSC** da chamada MQGET.. Essa correspondência é adicional a quaisquer outras correspondências que possam se aplicar (por exemplo, o identificador de correlação).

Se essa opção não for especificada, o campo *MDGID* no parâmetro **MSGDSC** será ignorado e qualquer identificador de grupo será correspondido.

**Nota:** O identificador de grupo GINONE é um valor especial que corresponde a qualquer identificador de grupo no MQMD da mensagem. Portanto, especificar MOGRPI com GINONE é o mesmo que não especificar MOGRPI.

### **MOSEQN**

Recuperar mensagem com o número de sequência da mensagem especificado.

Esta opção especifica que a mensagem a ser recuperada deve ter um número de sequência de mensagens que corresponda ao valor do campo *MDSEQ* no parâmetro **MSGDSC** da chamada MQGET. Essa correspondência é além de quaisquer outras correspondências que possam se aplicar (por exemplo, o identificador de grupo).

Se essa opção não for especificada, o campo *MDSEQ* no parâmetro **MSGDSC** será ignorado e qualquer número de sequência de mensagens corresponderá.

### **MOFFS**

Recuperar mensagem com deslocamento especificado.

Esta opção especifica que a mensagem a ser recuperada deve ter um deslocamento que corresponda ao valor do campo *MDOFF* no parâmetro **MSGDSC** da chamada MQGET. Essa correspondência é além de quaisquer outras correspondências que possam se aplicar (por exemplo, o número de sequência da mensagem).

Se essa opção não for especificada, o campo *MDOFF* no parâmetro **MSGDSC** será ignorado e quaisquer correspondências de deslocamento.

Se nenhuma das opções descritas for especificada, a seguinte opção poderá ser usada:

### **MONONE**

Nenhuma correspondência.

Essa opção especifica que nenhuma correspondência deve ser usada na seleção da mensagem a ser retornada; portanto, todas as mensagens na fila são elegíveis para recuperação (mas estão sujeitas ao controle pelas opções GMAMSA, GMASGA e GMCMPM)



MONONE é definido para a documentação do programa de auxílio Não se pretende que essa opção seja usada com qualquer outra opção MO\*, mas como seu valor é zero, tal uso não pode ser detectado.

Este campo é um campo de entrada O valor inicial desse campo é MOMSGI com MOCORI. Esse campo será ignorado se *GMVER* for menor que *GMVER2*.

**Nota:** O valor inicial do campo *GMMO* é definido para compatibilidade com gerenciadores de fila da versão anterior. No entanto, ao ler uma série de mensagens de uma fila sem usar critérios de seleção, esse valor inicial requer que o aplicativo reconfigure os campos *MDMID* e *MDCID* para MINONE e CINONE antes de cada chamada MQGET. A necessidade de reconfigurar *MDMID* e *MDCID* pode ser evitada configurando *GMVER* como *GMVER2* e *GMMO* como MONONE.

### **GMOPT (número inteiro assinado de 10 dígitos)**

Opções que controlam a ação de MQGET..

Zero ou mais das opções descritas a seguir podem ser especificadas: Se mais de um for necessário, os valores poderão ser adicionados (não inclua a mesma constante mais de uma vez). Combinações de opções inválidas são observadas; todas as outras combinações são válidas.

**Opções de Espera:** As opções a seguir estão relacionadas à espera de mensagens para chegar na fila:

#### **GMWT**

Aguarde a mensagem chegar.

O aplicativo deve aguardar até que uma mensagem adequada chegue.. O tempo máximo que o aplicativo aguarda é especificado em *GMWT*..

Se as solicitações MQGET forem inibidas ou as solicitações MQGET forem inibidas durante a espera, a espera será cancelada e a chamada será concluída com CCFAIL e o código de razão RC2016, independentemente de haver mensagens adequadas na fila.

Essa opção pode ser usada com as opções GMBRWF ou GMBRWN

Se vários aplicativos estiverem aguardando na mesma fila compartilhada, o aplicativo ou aplicativos que são ativados quando uma mensagem adequada chega serão descritos posteriormente nesta seção.

**Nota:** Na descrição a seguir, uma chamada MQGET de procura é aquela que especifica uma das opções de procura, mas não GMLK; uma chamada MQGET que especifica a opção GMLK é tratada como uma chamada não de procura

- Se uma ou mais chamadas MQGET não navegadas estiverem esperando, mas nenhuma chamada MQGET de navegação estiver aguardando, uma delas será ativada
- Se uma ou mais chamadas MQGET de procura estiverem aguardando, mas nenhuma chamada MQGET de procura estiver aguardando, todas serão ativadas..
- Se uma ou mais chamadas MQGET não navegadas e uma ou mais chamadas MQGET navegadas estiverem esperando, uma chamada MQGET não navegada será ativada, e nenhuma, algumas ou todas as chamadas MQGET navegadas... (O número de chamadas MQGET de procura ativadas não pode ser previsto, porque depende das considerações de planejamento do sistema operacional e de outros fatores.)

Se mais de uma chamada MQGET não de procura estiver aguardando na mesma fila, apenas uma será ativada; nessa situação, o gerenciador de filas tentará dar prioridade à espera de chamadas não de procura na seguinte ordem:

1. Solicitações get-wait específicas que podem ser atendidas apenas por determinadas mensagens, por exemplo, aquelas com um *MDMID* ou *MDCID* específico (ou ambos).
2. Solicitações get-wait gerais que podem ser satisfeitas por qualquer mensagem.

Os seguintes pontos devem ser observados:

- Na primeira categoria, nenhuma prioridade adicional é fornecida para solicitações get-wait mais específicas, por exemplo, aquelas que especificam *MDMID* e *MDCID*.

- Em qualquer categoria, não é possível prever qual aplicativo está selecionado. Em particular, o aplicativo que espera mais tempo não é necessariamente aquele selecionado.
- O comprimento do caminho e as considerações de planejamento de prioridade do sistema operacional podem significar que um aplicativo em espera de prioridade do sistema operacional inferior ao esperado recupera a mensagem.
- Também pode acontecer que um aplicativo que não está esperando recupere a mensagem em preferência a uma que esteja..

GMWT é ignorado se especificado com GMBRWC ou GMMUC; nenhum erro é gerado.

### **GMNWT**

Retornar imediatamente se nenhuma mensagem adequada.

O aplicativo não deve esperar se nenhuma mensagem adequada estiver disponível Este é o oposto da opção GMWT, e é definido para ajudar a documentação do programa. Ele será o padrão se nenhum for especificado

### **GMFIQ**

Falha se o gerenciador de filas estiver quiesce.

Essa opção força a chamada MQGET a falhar se o gerenciador de filas estiver em quiesce.

Se essa opção for especificada juntamente com GMWT e a espera estiver pendente no momento em que o gerenciador de filas entrar no estado quiesce:

- A espera é cancelada e a chamada retorna o código de conclusão CCFAIL com código de razão RC2161 .

Se GMFIQ não for especificado e o gerenciador de filas entrar no estado quiesce, a espera não será cancelada.

**Opções de Ponto de Sincronização:** As opções a seguir estão relacionadas à participação da chamada MQGET em uma unidade de trabalho:

### **GMSYP**

Obter mensagem com controle de ponto de sincronização.

A solicitação é operar dentro dos protocolos normais de unidade de trabalho. A mensagem é marcada como estando indisponível para outros aplicativos, mas é excluída da fila somente quando a unidade de trabalho é confirmada A mensagem é disponibilizada novamente se a unidade de trabalho for recuperada.

Se esta opção ou GMNSYP não for especificada, o pedido de obtenção não estará dentro de uma unidade de trabalho

Essa opção não é válida com nenhuma das opções a seguir:

- GMBRWF
- GMBRWC
- GMBRWN
- GMLK
- GMNSYP
- GMPSYP
- GMUNLK

### **GMPSYP**

Obter mensagem com controle de ponto de sincronização se mensagem for persistente.

A solicitação é operar dentro dos protocolos normais de unidade de trabalho, mas apenas se a mensagem recuperada for persistente Uma mensagem persistente possui o valor PEPER no campo *MDPER* em MQMD.

- Se a mensagem for persistente, o gerenciador de filas processará a chamada como se o aplicativo tivesse especificado GMSYP
- Se a mensagem não for persistente, o gerenciador de filas processará a chamada como se o aplicativo tivesse especificado GMNSYP (consulte a seção a seguir para obter detalhes.)

Essa opção não é válida com nenhuma das opções a seguir:

- GMBRWF
- GMBRWC
- GMBRWN
- GMCMPM
- GMNSYP
- GMSYP
- GMUNLK

### **GMNSYP**

Obter mensagem sem controle de ponto de sincronização.

O pedido é para operar fora dos protocolos normais de unidade de trabalho. A mensagem é excluída da fila imediatamente (a menos que seja uma solicitação de procura). A mensagem não pode ser disponibilizada novamente, efetuando backout da unidade de trabalho

Essa opção é assumida se GMBRWF ou GMBRWN for especificado

Se esta opção e GMSYP não forem especificados, o pedido de obtenção não estará dentro de uma unidade de trabalho

Essa opção não é válida com nenhuma das opções a seguir:

- GMSYP
- GMPSYP

**Procurar opções:** As opções a seguir estão relacionadas à procura de mensagens na fila:

### **GMBRWF**

Navegar a partir do início da fila.

Quando uma fila é aberta com a opção OOBW, um cursor de procura é estabelecido, posicionado logicamente antes da primeira mensagem na fila. Chamadas MQGET subsequentes que especificam a opção GMBRWF, GMBRWN ou GMBRWC podem ser usadas para recuperar mensagens da fila não destrutivamente. O cursor de procura marca a posição, nas mensagens na fila, a partir da qual a próxima chamada MQGET com GMBRWN procura uma mensagem adequada.

Uma chamada MQGET com GMBRWF faz a posição anterior do cursor de navegação ser ignorada. A primeira mensagem na fila que satisfaça as condições especificadas no descritor de mensagens é recuperada. A mensagem permanece na fila e o cursor de navegação está posicionado nesta mensagem.

Após essa chamada, o cursor de navegação é posicionado sobre a mensagem que foi retornada. Se a mensagem for removida da fila antes que a próxima chamada MQGET com GMBRWN seja emitida, o cursor de navegação permanecerá na posição na fila que a mensagem ocupou, mesmo que essa posição agora esteja vazia.

A opção GMMUC pode, então, ser usada com uma chamada MQGET não de navegação, se necessário, para remover a mensagem da fila.

O cursor de procura não é movido por uma chamada MQGET não de procura usando a mesma manipulação do *HOB*. Ele também não é movido por uma chamada MQGET de procura que retorna um código de conclusão de CCFAIL ou um código de razão de RC2080.

A opção GMLK pode ser especificada junto com esta opção, para fazer com que a mensagem que é procurada seja bloqueada.

GMBRWF pode ser especificado com qualquer combinação válida das opções GM\* e MO\* que controlam o processamento de mensagens em grupos e segmentos de mensagens lógicas.

Se GMLOGO for especificado, as mensagens serão procuradas em ordem lógica.. Se essa opção for omitida, as mensagens serão procuradas em ordem física. Quando GMBRWF é especificado, é possível alternar entre a ordem lógica e a ordem física, mas as chamadas MQGET subsequentes usando GMBRWN devem procurar a fila na mesma ordem que a chamada mais recente que especificou GMBRWF para a manipulação de fila

As informações do grupo e do segmento que o gerenciador de filas retém para chamadas MQGET que procuram mensagens na fila são separadas das informações do grupo e do segmento que o gerenciador de filas retém para chamadas MQGET que removem mensagens da fila. Quando o GMBRWF é especificado, o gerenciador de filas ignora as informações de grupo e de segmento para navegação e varre a fila como se não houvesse nenhum grupo atual e nenhuma mensagem lógica atual. Se a chamada MQGET for bem-sucedida (código de conclusão CCOK ou CCWARN), as informações do grupo e do segmento para navegação serão configuradas para aquela da mensagem retornada; se a chamada falhar, as informações do grupo e do segmento permanecerão as mesmas que antes da chamada.

Essa opção não é válida com nenhuma das opções a seguir:

- GMBRWC
- GMBRWN
- GMMUC
- GMSYP
- GMPSYP
- GMUNLK

Também será um erro se a fila não tiver sido aberta para procura

## **GMBRWN**

Procurar da posição atual na fila.

O cursor de navegação é avançado para a próxima mensagem na fila que satisfaça os critérios de seleção especificados na chamada MQGET. A mensagem é retornada ao aplicativo, mas permanece na fila.

Após uma fila ter sido aberta para procura, a primeira chamada de procura usando o identificador terá o mesmo efeito se especificar a opção GMBRWF ou GMBRWN.

Se a mensagem for removida da fila antes que a próxima chamada MQGET com GMBRWN seja emitida, o cursor de navegação permanecerá logicamente na posição na fila que a mensagem ocupou, mesmo que essa posição agora esteja vazia

As mensagens são armazenadas na fila de uma das duas maneiras:

- FIFO dentro da prioridade (MSPRIO), ou
- IFOP, independentemente da prioridade (MSFIFO)

O atributo da fila **MsgDeliverySequence** indica qual método se aplica (consulte [“Atributos para filas”](#) na página 1416 para obter detalhes).

Se a fila tiver um *MsgDeliverySequence* de MSPRIO e uma mensagem chegar na fila que é de uma prioridade mais alta do que aquela apontada atualmente pelo cursor de navegação, essa mensagem não será localizada durante a varredura atual da fila usando GMBRWN. Ele só pode ser localizado após o cursor de navegação ter sido reconfigurado com GMBRWF (ou reabrindo a fila).

A opção GMMUC pode ser usada posteriormente com uma chamada MQGET não de procura, se necessário, para remover a mensagem da fila.

O cursor de navegação não é movido pelas chamadas MQGET não navegadas usando a mesma manipulação *HOBJ*

A opção GMLK pode ser especificada junto com esta opção, para fazer com que a mensagem que é procurada seja bloqueada

GMBRWN pode ser especificado com qualquer combinação válida das opções GM\* e MO\* que controlam o processamento de mensagens em grupos e segmentos de mensagens lógicas.

Se GMLOGO for especificado, as mensagens serão procuradas em ordem lógica.. Se essa opção for omitida, as mensagens serão procuradas em ordem física Quando GMBRWF é especificado, é possível alternar entre a ordem lógica e a ordem física, mas as chamadas MQGET subsequentes usando GMBRWN devem procurar a fila na mesma ordem que a chamada mais recente que especificou GMBRWF para a manipulação de fila A chamada falha com o código de razão RC2259 se essa condição não for satisfeita.

**Nota:** É necessário cuidado especial se uma chamada MQGET for usada para navegar além do final de um grupo de mensagens (ou mensagem lógica não em um grupo) quando GMLOGO não for especificado. Por exemplo, se a última mensagem no grupo preceder a primeira mensagem no grupo na fila, usando GMBRWN para navegar além do final do grupo, especificar MOSEQN com MDSEQ configurado como 1 (para localizar a primeira mensagem do próximo grupo) retornará novamente a primeira mensagem no grupo já procurada. Isso pode acontecer imediatamente ou um número de chamadas MQGET posterior (se houver grupos de intervenção).

A possibilidade de um loop infinito pode ser evitada abrindo a fila duas vezes para navegar:

- Use o primeiro identificador para pesquisar apenas a primeira mensagem em cada grupo.
- Use o segundo identificador para pesquisar apenas as mensagens em um grupo específico.
- Use as opções MO\* para mover o segundo cursor de navegação para a posição do primeiro cursor de navegação, antes de procurar as mensagens no grupo.
- Não use GMBRWN para navegar além do fim de um grupo.

As informações de grupo e de segmento que o gerenciador de filas retém para chamadas MQGET que procuram mensagens na fila são separadas das informações de grupo e de segmento que elas retêm para chamadas MQGET que removem mensagens da fila.

Essa opção não é válida com nenhuma das opções a seguir:

- GMBRWF
- GMBRWC
- GMMUC
- GMSYP
- GMPSYP
- GMUNLK

Também será um erro se a fila não tiver sido aberta para procura

## **GMBRWC**

Navegar pela mensagem sob o cursor de navegação.

Essa opção faz com que a mensagem apontada pelo cursor de navegação seja recuperada não destrutivamente, independentemente das opções MO\* especificadas no campo GMMO em MQGMO.

A mensagem apontada pelo cursor de navegação é aquela que foi recuperada por último utilizando a opção GMBRWF ou GMBRWN. A chamada falhará se nenhuma dessas chamadas tiver sido emitida para essa fila desde que foi aberta ou se a mensagem que estava sob o cursor de navegação tiver sido recuperada destrutivamente.

A posição do cursor de navegação não é alterada por esta chamada..

A opção GMMUC pode, então, ser usada com uma chamada MQGET não de navegação, se necessário, para remover a mensagem da fila.

O cursor de procura não é movido por uma chamada MQGET não de procura usando a mesma manipulação do *HOB*J Também não é movido por uma chamada MQGET de procura que retorna um código de conclusão CCFAIL ou um código de razão de RC2080.

Se GMBRWC for especificado com GMLK:

- Se já houver uma mensagem bloqueada, ela deverá ser aquela sob o cursor, para que seja retornada sem desbloqueio e realocada; a mensagem permanece bloqueada.
- Se não houver mensagem bloqueada, a mensagem sob o cursor de navegação (se houver uma) será bloqueada e retornada ao aplicativo; se não houver mensagem sob o cursor de navegação, a chamada falhará.

Se GMBRWC for especificado sem GMLK:

- Se já houver uma mensagem bloqueada, ela deverá ser aquela abaixo do cursor. Essa mensagem é retornada para o aplicativo e, em seguida, desbloqueada. Como a mensagem agora está desbloqueada, não há garantia de que ela possa ser procurada novamente ou recuperada destrutivamente (ela pode ser recuperada destrutivamente por outro aplicativo obtendo mensagens da fila).
- Se não houver nenhuma mensagem bloqueada, a mensagem sob o cursor de navegação (se houver uma) será retornada para o aplicativo; se não houver nenhuma mensagem sob o cursor de navegação, a chamada falhará.

Se GMCMPM for especificado com GMBRWC, o cursor de procura deverá identificar uma mensagem com um campo *MDOFF* em MQMD que seja zero. Se essa condição não for atendida, a chamada falhará com o código de razão RC2246

As informações de grupo e de segmento que o gerenciador de filas retém para chamadas MQGET que procuram mensagens na fila são separadas das informações de grupo e de segmento que elas retém para chamadas MQGET que removem mensagens da fila.

Essa opção não é válida com nenhuma das opções a seguir:

- GMBRWF
- GMBRWN
- GMMUC
- GMSYP
- GMPSYP
- GMUNLK

Também será um erro se a fila não tiver sido aberta para procura

### **GMMUC**

Obter mensagem sob cursor de navegação.

Essa opção faz com que a mensagem apontada pelo cursor de navegação seja recuperada, independentemente das opções MO\* especificadas no campo *GMMO* no MQGMO. A mensagem é removida da fila.

A mensagem apontada pelo cursor de navegação é aquela que foi recuperada por último utilizando a opção GMBRWF ou GMBRWN.

Se GMCMPM for especificado com GMMUC, o cursor de procura deverá identificar uma mensagem com um campo *MDOFF* em MQMD que seja zero. Se essa condição não for atendida, a chamada falhará com o código de razão RC2246

Essa opção não é válida com nenhuma das opções a seguir:

- GMBRWF
- GMBRWC
- GMBRWN
- GMUNLK

Também será um erro se a fila não tiver sido aberta para procura e para entrada. Se o cursor de navegação não estiver apontando atualmente para uma mensagem recuperável, um erro será retornado pela chamada MQGET.

**Opções de bloqueio:** As seguintes opções estão relacionadas a mensagens de bloqueio na fila:

### **GMLK**

Bloquear mensagem.

Essa opção bloqueia a mensagem que é procurada, para que a mensagem fique invisível para qualquer outro identificador aberto para a fila. A opção pode ser especificada apenas se uma das seguintes opções também for especificada:

- GMBRWF
- GMBRWN
- GMBRWC

Apenas uma mensagem pode ser bloqueada por identificador de fila, mas esta pode ser uma mensagem lógica ou uma mensagem física:

- Se GMCMPM for especificado, todos os segmentos de mensagens que compõem a mensagem lógica serão bloqueados para o identificador de filas (se todos estiverem presentes na fila e disponíveis para recuperação)..
- Se GMCMPM não for especificado, apenas uma única mensagem física será bloqueada no identificador de filas. Se essa mensagem for um segmento de uma mensagem lógica, o segmento bloqueado evitará que outros aplicativos usem o GMCMPM para recuperar ou procurar a mensagem lógica.

A mensagem bloqueada é sempre aquela sob o cursor de procura e a mensagem pode ser removida da fila por uma chamada MQGET posterior que especifica a opção GMMUC. Outras chamadas MQGET usando o identificador de filas também podem remover a mensagem (por exemplo, uma chamada que especifica o identificador de mensagem da mensagem bloqueada).

Se a chamada retornar o código de conclusão CCFAIL ou CCWARN com código de razão RC2080, nenhuma mensagem será bloqueada.

Se o aplicativo decidir não remover a mensagem da fila, o bloqueio será liberado por:

- Emitir outra chamada MQGET para essa manipulação, com GMBRWF ou GMBRWN especificado (com ou sem GMLK); a mensagem será desbloqueada se a chamada for concluída com CCOK ou CCWARN, mas permanecerá bloqueada se a chamada for concluída com CCFAIL. No entanto, as exceções a seguir se aplicam:
  - A mensagem não será desbloqueada se CCWARN for retornado com RC2080..
  - A mensagem será desbloqueada se CCFAIL for retornado com RC2033..

Se GMLK também for especificado, a mensagem retornada será bloqueada.. Se GMLK não for especificado, não haverá nenhuma mensagem bloqueada após a chamada.

Se GMWT for especificado e nenhuma mensagem estiver imediatamente disponível, o desbloqueio na mensagem original ocorrerá antes do início da espera (desde que a chamada esteja livre de erros).

- Emitir outra chamada MQGET para essa manipulação, com GMBRWC (sem GMLK); a mensagem será desbloqueada se a chamada for concluída com CCOK ou CCWARN, mas permanecerá bloqueada se a chamada for concluída com CCFAIL. No entanto, a exceção a seguir se aplica:
  - A mensagem não será desbloqueada se CCWARN for retornado com RC2080..
- Emitindo outra chamada MQGET para essa manipulação com GMUNLK
- Emitir uma chamada MQCLOSE para esse identificador (explícita ou implicitamente pelo término do aplicativo).

Nenhuma opção de abertura especial é necessária para especificar essa opção, diferente de OOBROW, que é necessária para especificar a opção de navegação associada.

Essa opção não é válida com nenhuma das opções a seguir:

- GMSYP
- GMPSYP
- GMUNLK

#### **GMUNLK**

Desbloquear mensagem.

A mensagem a ser desbloqueada deve ter sido bloqueada anteriormente por uma chamada MQGET com a opção GMLK. Se não houver nenhuma mensagem bloqueada para essa manipulação, a chamada será concluída com CCWARN e RC2209.

Os parâmetros **MSGDSC**, **BUFLN**, **BUFFER** e **DATLEN** não serão verificados ou alterados se GMUNLK for especificado.. Nenhuma mensagem é retornada em *BUFFER*.

Nenhuma opção de abertura especial é necessária para especificar essa opção (embora o OOBROW seja necessário para emitir o pedido de bloqueio em primeiro lugar).

Essa opção não é válida com nenhuma opção, exceto a seguinte:

- GMNWT
- GMNSYP

Ambas as opções são assumidas se especificadas ou não.

**Opções de dados da mensagem:** as opções a seguir estão relacionadas ao processamento dos dados da mensagem quando a mensagem é lida a partir da fila:

#### **GMATM**

Permitir truncamento dos dados da mensagem.

Se o buffer de mensagem for muito pequeno para conter a mensagem completa, essa opção permitirá que a chamada MQGET preencha o buffer com o máximo da mensagem que o buffer pode conter. Isso significa:

- Ao navegar pelas mensagens, o cursor de navegação é avançado para a mensagem retornada.
- Ao remover mensagens, a mensagem retornada é removida da fila.
- O código de razão RC2079 será retornado se não ocorrer nenhum outro erro.

Sem essa opção, o buffer ainda é preenchido com a maior parte da mensagem que ele pode conter, um código de conclusão de aviso é emitido, mas o processamento não é concluído. Isso significa:

- Ao navegar pelas mensagens, o cursor de navegação não é avançado.
- Ao remover mensagens, a mensagem não é removida da fila..
- O código de razão RC2080 será retornado se não ocorrer nenhum outro erro.

#### **GMCONV**

Converter dados da mensagem.

Essa opção solicita que os dados do aplicativo na mensagem sejam convertidos em conformidade com os valores *MDCSI* e *MDENC* especificados no parâmetro **MSGDSC** na chamada MQGET, antes que os dados sejam copiados para o parâmetro **BUFFER**.

O campo *MDFMT* especificado quando a mensagem foi colocada é assumido pelo processo de conversão para identificar a natureza dos dados na mensagem. A conversão dos dados da mensagem é pelo gerenciador de filas para formatos integrados e por uma saída gravada pelo usuário para outros formatos.

- Se a conversão for executada com êxito, os campos *MDCSI* e *MDENC* especificados no parâmetro **MSGDSC** permanecerão inalterados no retorno da chamada MQGET.
- Se a conversão não puder ser executada com êxito (mas a chamada MQGET for concluída sem erro), os dados da mensagem serão retornados sem conversão e os campos *MDCSI* e *MDENC*



em *MSGDSC* serão configurados para os valores para a mensagem não convertida. Neste caso, o código de conclusão é *CCWARN*

Em qualquer caso, portanto, esses campos descrevem o identificador de conjunto de caracteres e a codificação dos dados da mensagem que são retornados no parâmetro **BUFFER**.

Consulte o campo *MDFMT* descrito em “*MQMD (Descritor de mensagens) em IBM i*” na página 1148 para obter uma lista de nomes de formato para os quais o gerenciador de fila executa a conversão

**Opções de grupo e segmento:** As opções a seguir estão relacionadas ao processamento de mensagens em grupos e segmentos de mensagens lógicas. Essas definições podem ajudar a entender as opções:

#### **Mensagem física**

Esta é a menor unidade de informações que pode ser colocada ou removida de uma fila; geralmente corresponde às informações especificadas ou recuperadas em uma chamada *MQPUT*, *MQPUT1* ou *MQGET*. Cada mensagem física possui seu próprio descritor de mensagens (*MQMD*). Geralmente, as mensagens físicas são distinguidas por valores diferentes para o identificador de mensagens (campo *MDMID* em *MQMD*), embora isso não seja aplicado pelo gerenciador de filas.

#### **Mensagem lógica**

Esta é uma única unidade de informações do aplicativo. Na ausência de restrições do sistema, uma mensagem lógica seria a mesma que uma mensagem física. Mas, quando as mensagens lógicas são grandes, as restrições do sistema podem tornar aconselhável ou necessário dividir uma mensagem lógica em duas ou mais mensagens físicas, chamadas segmentos.

Uma mensagem lógica que foi segmentada consiste em duas ou mais mensagens físicas que possuem o mesmo identificador de grupo não nulo (campo *MDGID* em *MQMD*) e o mesmo número de sequência de mensagens (campo *MDSEQ* em *MQMD*). Os segmentos são diferenciados por valores diferentes para o deslocamento de segmento (campo *MDOFF* em *MQMD*), que fornece o deslocamento dos dados na mensagem física do início dos dados na mensagem lógica. Como cada segmento é uma mensagem física, os segmentos em uma mensagem lógica geralmente possuem identificadores de mensagem diferentes..

Uma mensagem lógica que não foi segmentada, mas para a qual a segmentação foi permitida pelo aplicativo de envio, também possui um identificador de grupo não nulo, embora neste caso haja apenas uma mensagem física com esse identificador de grupo se a mensagem lógica não pertencer a um grupo de mensagens. Mensagens lógicas para as quais a segmentação foi inibida pelo aplicativo de envio possuem um identificador de grupo nulo (*GINONE*), a menos que a mensagem lógica pertença a um grupo de mensagens.

#### **Grupo de mensagens**

Este é um conjunto de uma ou mais mensagens lógicas que possuem o mesmo identificador de grupo não nulo.. As mensagens lógicas no grupo são distinguidas por valores diferentes para o número de sequência da mensagem, que é um número inteiro no intervalo de 1 a *n*, em que *n* é o número de mensagens lógicas no grupo. Se uma ou mais mensagens lógicas forem segmentadas, haverá mais de *n* mensagens físicas no grupo.

#### **GMLOGO**

As mensagens em grupos e segmentos de mensagens lógicas são retornadas em ordem lógica.

Esta opção controla a ordem na qual as mensagens são retornadas por chamadas *MQGET* sucessivas para o identificador de fila. A opção deve ser especificada em cada uma dessas chamadas para ter efeito.

Se *GMLOGO* for especificado para chamadas *MQGET* sucessivas para o identificador de fila, as mensagens em grupos serão retornadas na ordem fornecida por seus números de sequência de mensagem e os segmentos de mensagens lógicas serão retornadas na ordem fornecida por suas compensações de segmentos... Essa ordem pode ser diferente da ordem na qual essas mensagens e segmentos ocorrem na fila..

**Nota:** Especificar *GMLOGO* não tem consequências adversas em mensagens que não pertencem a grupos e que não são segmentos. Com efeito, essas mensagens são tratadas como se cada

uma pertencesse a um grupo de mensagens que consiste em apenas uma mensagem. Portanto, é perfeitamente seguro especificar GMLOGO ao recuperar mensagens de filas que podem conter uma mistura de mensagens em grupos, segmentos de mensagens e mensagens não segmentados não em grupos.

Para retornar as mensagens na ordem necessária, o gerenciador de filas retém as informações de grupo e segmento entre chamadas MQGET sucessivas. Essas informações identificam o grupo de mensagens atual e a mensagem lógica atual para o identificador de filas, a posição atual no grupo e a mensagem lógica e se as mensagens estão sendo recuperadas em uma unidade de trabalho. Como o gerenciador de filas retém essas informações, o aplicativo não precisa configurar as informações do grupo e do segmento antes de cada chamada MQGET. Especificamente, isso significa que o aplicativo não precisa configurar os campos *MDGID*, *MDSEQ* e *MDOFF* no MQMD. No entanto, o aplicativo não precisa configurar a opção GMSYP ou GMNSYP corretamente em cada chamada.

Quando a fila é aberta não há grupo de mensagens atual e nenhuma mensagem lógica atual. Um grupo de mensagens se torna o grupo de mensagens atual quando uma mensagem que possui a sinalização MFMIG é retornada pela chamada MQGET. Com GMLOGO especificado em chamadas sucessivas, esse grupo permanece o grupo atual até que seja retornada uma mensagem que tenha:

- MFLMIG sem MFSEG (ou seja, a última mensagem lógica no grupo não é segmentada) ou
- MFLMIG com MFLSEG (ou seja, a mensagem retornada é o último segmento da última mensagem lógica no grupo)

Quando essa mensagem é retornada, o grupo de mensagens é finalizado e, na conclusão bem-sucedida dessa chamada MQGET, não há mais um grupo atual. De maneira semelhante, uma mensagem lógica torna-se a mensagem lógica atual quando uma mensagem que possui o sinalizador MFSEG é retornada pela chamada MQGET e essa mensagem lógica é finalizada quando a mensagem que possui o sinalizador MFLSEG é retornada.

Se nenhum critério de seleção for especificado, chamadas MQGET sucessivas retornarão (na ordem correta) as mensagens para o primeiro grupo de mensagens na fila, então as mensagens para o segundo grupo de mensagens e assim por diante, até que não haja mais mensagens disponíveis. É possível selecionar os grupos de mensagens específicos retornados especificando uma ou mais das seguintes opções no campo *GMMO* :

- MOMSGI
- MOCORI
- MOGRPI

No entanto, essas opções são efetivas apenas quando não há nenhum grupo de mensagens ou mensagem lógica atual; consulte o campo *GMMO* descrito neste tópico.

Tabela 702 na página 1127 mostra os valores dos campos *MDMID*, *MDCID*, *MDGID*, *MDSEQ* e *MDOFF* que o gerenciador de filas procura ao tentar localizar uma mensagem para retornar na chamada MQGET. Isso se aplica para remover mensagens da fila e procurar mensagens na fila. As colunas na tabela têm os significados a seguir:

**LOG ORD**

Indica se a opção GMLOGO é especificada na chamada.

**Cur grp**

Indica se um grupo de mensagens atual existe antes da chamada.

**Cur log msg**

Indica se uma mensagem lógica atual existe antes da chamada.

**Outras colunas**

Mostrar os valores que o Gerenciador de Filas procura. "Anterior" indica o valor retornado para o campo na mensagem anterior para o identificador de filas.

Tabela 702. Opções MQGET relacionadas a mensagens em grupos e segmentos de mensagens lógicas

Opções que você especifica	Grupo e status log-msg antes da chamada		Valores que o gerenciador de filas procura				
	LOG ORD	Cur grp	Cur log msg	MDMID	MDCID	MDGID	MDSEQ
Sim	No	No	Controlado por GMMO	Controlado por GMMO	Controlado por GMMO	1	0
Sim	No	Sim	Qualquer identificador de mensagem	Qualquer identificador de correlação	Identificador do grupo anterior	1	Deslocamento anterior + comprimento do segmento anterior
Sim	Sim	No	Qualquer identificador de mensagem	Qualquer identificador de correlação	Identificador do grupo anterior	Número da sequência anterior + 1	0
Sim	Sim	Sim	Qualquer identificador de mensagem	Qualquer identificador de correlação	Identificador do grupo anterior	Número de sequência anterior	Deslocamento anterior + comprimento do segmento anterior
No	Qualquer um	Qualquer um	Controlado por GMMO	Controlado por GMMO	Controlado por GMMO	Controlado por GMMO	Controlado por GMMO

Quando diversos grupos de mensagens estão presentes na fila e são elegíveis para retorno, os grupos são retornados na ordem determinada pela posição na fila do primeiro segmento da primeira mensagem lógica em cada grupo (ou seja, as mensagens físicas que possuem números de sequência de mensagens de 1 e deslocamentos de 0, determinam a ordem na qual os grupos elegíveis são retornados).

A opção GMLOGO afeta unidades de trabalho da seguinte forma:

- Se a primeira mensagem lógica ou segmento em um grupo for recuperado em uma unidade de trabalho, todas as outras mensagens lógicas e segmentos no grupo deverão ser recuperados em uma unidade de trabalho, se o mesmo identificador de fila for usado. No entanto, eles não precisam ser recuperados dentro da mesma unidade de trabalho. Isso permite que um grupo de mensagens que consiste em muitas mensagens físicas seja dividido em duas ou mais unidades consecutivas de trabalho para o identificador da fila.
- Se a primeira mensagem lógica ou segmento em um grupo não for recuperado dentro de uma unidade de trabalho, nenhuma das outras mensagens lógicas e segmentos no grupo poderão ser recuperados dentro de uma unidade de trabalho, se a mesma manipulação de filas for utilizada...

Se essas condições não forem atendidas, a chamada MQGET falhará com o código de razão RC2245

Quando GMLOGO é especificado, o MQGMO fornecido na chamada MQGET não deve ser menor que GMVER2e o MQMD não deve ser menor que MDVER2. Se essa condição não for atendida, a chamada falhará com o código de razão RC2256 ou RC2257, conforme apropriado

Se GMLOGO não for especificado para chamadas MQGET sucessivas para o identificador de fila, as mensagens serão retornadas sem considerar se pertencem a grupos de mensagens ou se são segmentos de mensagens lógicas. Isso significa que as mensagens ou segmentos de

um determinado grupo ou mensagem lógica podem ser retornados fora de ordem ou podem ser misturados com mensagens ou segmentos de outros grupos ou mensagens lógicas ou com mensagens que não estão em grupos e não são segmentos. Nessa situação, as mensagens específicas retornadas por chamadas MQGET sucessivas são controladas pelas opções MO\* especificadas nessas chamadas (consulte o campo *GMMO* descrito em “MQGMO (opções GetMessage) em IBM i” na página 1113 para obter detalhes dessas opções).

Esta é a técnica que pode ser utilizada para reiniciar um grupo de mensagens ou mensagem lógica no meio, após uma falha do sistema ter ocorrido. Quando o sistema reiniciar, o aplicativo poderá configurar os campos *MDGID*, *MDSEQ*, *MDOFF* e *GMMO* para os valores apropriados e, em seguida, emitir a chamada MQGET com *GMSYP* ou *GMNSYP* configurado conforme necessário, mas sem especificar *GMLOGO*. Se essa chamada for bem-sucedida, o gerenciador de fila reterá as informações de grupo e de segmento e as chamadas MQGET subsequentes usando essa manipulação de fila poderão especificar *GMLOGO* normalmente.

As informações do grupo e do segmento que o gerenciador de filas retém para a chamada MQGET são separadas das informações do grupo e do segmento que ele retém para a chamada MQPUT. Além disso, o gerenciador de filas retém informações separadas para:

- Chamadas MQGET que removem mensagens da fila
- Chamadas MQGET que procuram mensagens na fila.

Para qualquer identificador de fila especificado, o aplicativo é livre para combinar chamadas MQGET que especificam *GMLOGO* com chamadas MQGET que não o fazem, mas os pontos a seguir devem ser observados:

- Se *GMLOGO* não for especificado, cada chamada MQGET bem-sucedida fará com que o gerenciador de filas configure as informações de grupo e segmento salvas para os valores correspondentes à mensagem retornada; isso substitui as informações de grupo e segmento existentes retidas pelo gerenciador de filas para o identificador de filas. Apenas as informações apropriadas para a ação da chamada (procurar ou remover) são modificadas.
- Se *GMLOGO* não for especificado, a chamada não falhará se houver um grupo de mensagens atual ou uma mensagem lógica; a chamada poderá, no entanto, ser bem-sucedida com um código de conclusão *CCWARN*. O [Tabela 703 na página 1128](#) mostra os diferentes casos que podem surgir. Nesses casos, se o código de conclusão não for *CCOK*, o código de razão será um dos seguintes:
  - RC2241
  - RC2242
  - RC2245

**Nota:** O gerenciador de filas não verifica as informações do grupo e do segmento ao navegar em uma fila ou ao fechar uma fila que foi aberta para navegação, mas não entrada; nesses casos, o código de conclusão é sempre *CCOK* (assumindo nenhum outro erro).

*Tabela 703. Resultado quando a chamada MQGET ou MQCLOSE não é consistente com informações de grupo e segmento*

<b>A chamada atual é</b>	<b>Chamada anterior era MQGET com GMLOGO</b>	<b>Chamada anterior era MQGET sem GMLOGO</b>
MQGET com GMLOGO	CCFAIL	CCFAIL
MQGET sem GMLOGO	CCWARN	CCOK
MQCLOSE com um grupo ou mensagem lógica não terminada	CCWARN	CCOK

Os aplicativos que simplesmente desejam recuperar mensagens e segmentos em ordem lógica são recomendados para especificar *GMLOGO*, já que esta é a opção mais simples de usar. Esta

opção livra o aplicativo da necessidade de gerenciar as informações de grupo e segmento, pois o gerenciador de filas gerencia essa informação. No entanto, os aplicativos especializados podem precisar de mais controle do que o fornecido pela opção GMLOGO e isso pode ser obtido não especificando essa opção. Se isso for feito o aplicativo deve assegurar que os campos *MDMID*, *MDCID*, *MDGID*, *MDSEQ* e *MDOFF* no MQMD, e as opções MO\* no GMMO no MQGMO, sejam configurados corretamente, antes de cada chamada MQGET.

Por exemplo, um aplicativo que deseja encaminhar mensagens físicas que ele recebe, sem considerar se essas mensagens estão em grupos ou segmentos de mensagens lógicas, não deve especificar GMLOGO. Isso ocorre porque em uma rede complexa com vários caminhos entre o envio e o recebimento de gerenciadores de filas, as mensagens físicas podem chegar fora de ordem. Ao não especificar GMLOGO e o PMLOGO correspondente na chamada MQPUT, o aplicativo de encaminhamento pode recuperar e encaminhar cada mensagem física assim que ela chegar, sem ter que esperar a chegada da próxima na ordem lógica...

GMLOGO pode ser especificado com qualquer uma das outras opções GM\*, e com várias das opções MO\* em circunstâncias apropriadas.

### GMCMPPM

Apenas mensagens lógicas completas podem ser recuperadas

Esta opção especifica que apenas uma mensagem lógica completa pode ser retornada por uma chamada MQGET. Se a mensagem lógica for segmentada, o gerenciador de filas remonta os segmentos e retorna a mensagem lógica completa para o aplicativo; o fato de que a mensagem lógica foi segmentada não é aparente para o aplicativo que a recupera.

**Nota:** Esta é a única opção que faz com que o gerenciador de filas remonte segmentos de mensagens. Se não especificado, os segmentos serão retornados individualmente para o aplicativo se estiverem presentes na fila (e eles satisfizerem os outros critérios de seleção especificados na chamada MQGET). Portanto, os aplicativos que não desejam receber segmentos individuais devem sempre especificar GMCMPPM.

Para usar essa opção, o aplicativo deve fornecer um buffer grande o suficiente para acomodar a mensagem completa ou especificar a opção GMATM.

Se a fila contiver mensagens segmentadas com alguns segmentos ausentes (talvez porque eles foram atrasados na rede e ainda não chegaram), especificar GMCMPPM impede a recuperação de segmentos pertencentes a mensagens lógicas incompletas. No entanto, esses segmentos de mensagem ainda contribuem para o valor do atributo da fila **CurrentQDepth**; isso significa que pode não haver mensagens lógicas recuperáveis, embora *CurrentQDepth* seja maior que zero.

Para mensagens persistentes, o gerenciador de filas pode remontar os segmentos apenas dentro de uma unidade de trabalho:

- Se a chamada MQGET estiver operando dentro de uma unidade de trabalho definida pelo usuário, essa unidade de trabalho será usada. Se a chamada falhar por meio do processo de remontagem, o gerenciador de filas restabelecerá na fila quaisquer segmentos que foram removidos durante a remontagem. Entretanto, a falha não impede que a unidade de trabalho seja confirmada com êxito.
- Se a chamada estiver operando fora de uma unidade de trabalho definida pelo usuário e não houver nenhuma unidade de trabalho definida pelo usuário existente, o gerenciador de filas criará uma unidade de trabalho apenas pela duração da chamada. Se a chamada for bem-sucedida, o gerenciador de fila confirma a unidade de trabalho automaticamente (o aplicativo não precisa fazer isso). Se a chamada falhar, o gerenciador de filas volta a unidade de trabalho.
- Se a chamada estiver operando fora de uma unidade de trabalho definida pelo usuário, mas uma unidade de trabalho definida pelo usuário existir, o gerenciador de filas não poderá executar a remontagem. Se a mensagem não precisar de remontagem, a chamada ainda poderá ser bem-sucedida. Mas se a mensagem requerer remontagem, a chamada falhará com o código de razão RC2255.

Para mensagens não persistentes, o gerenciador de filas não requer que uma unidade de trabalho esteja disponível para executar a remontagem.

Cada mensagem física que é um segmento possui seu próprio descritor de mensagens. Para os segmentos que constituem uma única mensagem lógica, a maioria dos campos no descritor de mensagens é a mesma para todos os segmentos na mensagem lógica—geralmente são apenas os campos *MDMID*, *MDOFF* e *MDMFL* que diferem entre segmentos na mensagem lógica. No entanto, se um segmento for colocado em uma fila de mensagens não entregues em um gerenciador de filas intermediário, o manipulador de fila de mensagens não entregues recuperará a mensagem especificando a opção *GMCONV* e isso poderá resultar no conjunto de caracteres ou na codificação do segmento sendo alterado. Se o manipulador DLQ enviar com êxito o segmento em seu caminho, o segmento poderá ter um conjunto de caracteres ou codificação que difere dos outros segmentos na mensagem lógica quando o segmento finalmente chegar ao gerenciador de filas de destino.

Uma mensagem lógica que consiste em segmentos nos quais o *MDCSI*, *MDENC* ou ambos os campos diferem não pode ser remontado pelo gerenciador de filas em uma única mensagem lógica. Em vez disso, o gerenciador de filas remonta e retorna os primeiros segmentos consecutivos no início da mensagem lógica que possuem os mesmos identificadores e codificações de conjunto de caracteres e a chamada *MQGET* é concluída com o código de conclusão *CCWARN* e o código de razão *RC2243* ou *RC2244*, conforme apropriado. Isso acontece independentemente de *GMCONV* ser especificado. Para recuperar os segmentos restantes, o aplicativo deve emitir novamente a chamada *MQGET* sem a opção *GMCPMP*, recuperando os segmentos um por um.. *GMLOGO* pode ser usado para recuperar os segmentos restantes em ordem.

Também é possível para um aplicativo que coloca segmentos para configurar outros campos no descritor de mensagem para valores que diferem entre segmentos. No entanto, não há vantagem em fazer isso se o aplicativo de recebimento usar o *GMCPMP* para recuperar a mensagem lógica. Quando o gerenciador de filas remonta uma mensagem lógica, ele retorna no descritor de mensagens os valores do descritor de mensagem para o primeiro segmento; a única exceção é o campo *MDMFL*, que o gerenciador de filas configura para indicar que a mensagem remontada é o único segmento.

Se *GMCPMP* for especificado para uma mensagem de relatório, o gerenciador de filas executará processamento especial. O gerenciador de filas verifica a fila para ver se todas as mensagens de relatório desse tipo de relatório relacionadas aos diferentes segmentos na mensagem lógica estão presentes na fila.. Se estiverem, eles poderão ser recuperados como uma única mensagem, especificando *GMCPMP*. Para que isso seja possível, as mensagens de relatório devem ser geradas por um gerenciador de filas ou *MCA* que suporta segmentação ou o aplicativo de origem deve solicitar pelo menos 100 bytes de dados da mensagem (ou seja, as opções *RO\* D* ou *RO\* F* apropriadas devem ser especificadas). Se a quantidade total de dados do aplicativo estiver presente para um segmento, os bytes ausentes serão substituídos por nulos na mensagem de relatório retornada.

Se *GMCPMP* for especificado com *GMMUC* ou *GMBRWC*, o cursor de navegação deverá ser posicionado em uma mensagem com um campo *MDOFF* em *MQMD* que tenha um valor 0. Se essa condição não for atendida, a chamada falhará com o código de razão *RC2246*.

*GMCPMP* implica *GMASGA*, que não precisa, portanto, ser especificado.

*GMCPMP* pode ser especificado com qualquer uma das outras opções *GM\** além de *GMPSYP*, e com qualquer uma das opções *MO\** além de *MOOFFS*.

## **GMAMSA**

Todas as mensagens no grupo devem ser disponibilizadas

Esta opção específica que as mensagens em um grupo se tornam disponíveis para recuperação apenas quando todas as mensagens no grupo estão disponíveis. Se a fila contiver grupos de mensagens com algumas das mensagens ausentes (talvez porque elas foram atrasadas na rede e ainda não chegaram), especificar *GMAMSA* evitará a recuperação de mensagens pertencentes a grupos incompletos. No entanto, essas mensagens ainda contribuem para o valor do atributo de fila **CurrentQDepth**; isso significa que pode não haver grupos de mensagens recuperáveis, embora **CurrentQDepth** seja maior que zero. Se não houver nenhuma outra mensagem

recuperável, o código de razão RC2033 será retornado após o intervalo de espera especificado (se houver) ter expirado.

O processamento de GMAMSA depende se GMLOGO também é especificado:

- Se ambas as opções forem especificadas, o GMAMSA afetará apenas quando não houver nenhum grupo atual ou mensagem lógica. Se houver um grupo atual ou mensagem lógica, GMAMSA será ignorado. Isso significa que o GMAMSA pode permanecer no processamento de mensagens em ordem lógica.
- Se GMAMSA for especificado sem GMLOGO, GMAMSA sempre terá um efeito. Isso significa que a opção deve ser desativada após a primeira mensagem no grupo ter sido removida da fila, para poder remover as mensagens restantes no grupo.

Conclusão bem-sucedida de uma chamada MQGET especificando GMAMSA significa que no momento em que a chamada MQGET foi emitida, todas as mensagens no grupo estavam na fila. No entanto, esteja ciente de que outros aplicativos ainda podem remover mensagens do grupo (o grupo não está bloqueado para o aplicativo que recupera a primeira mensagem no grupo).

Se essa opção não for especificada, as mensagens pertencentes a grupos poderão ser recuperadas, mesmo quando o grupo estiver incompleto.

GMAMSA implica GMASGA, que não precisa, portanto, ser especificado.

GMAMSA pode ser especificado com qualquer uma das outras opções GM\* e com qualquer uma das opções MO\*.

## **GMASGA**

Todos os segmentos em uma mensagem lógica devem estar disponíveis.

Esta opção especifica que os segmentos em uma mensagem lógica se tornam disponíveis para recuperação apenas quando todos os segmentos na mensagem lógica estão disponíveis... Se a fila contiver mensagens segmentadas com alguns dos segmentos ausentes (talvez porque eles foram atrasados na rede e ainda não chegaram), especificar GMASGA evitará a recuperação de segmentos pertencentes a mensagens lógicas incompletas. No entanto, esses segmentos ainda contribuem para o valor do atributo da fila **CurrentQDepth**; isso significa que pode não haver mensagens lógicas recuperáveis, embora **CurrentQDepth** seja maior que zero. Se não houver nenhuma outra mensagem recuperável, o código de razão RC2033 será retornado após o intervalo de espera especificado (se houver) ter expirado.

O processamento de GMASGA depende se GMLOGO também é especificado:

- Se ambas as opções forem especificadas, GMASGA terá efeito apenas quando não houver nenhuma mensagem lógica atual. Se houver uma mensagem lógica atual, o GMASGA será ignorado. Isso significa que o GMASGA pode permanecer ligado ao processar mensagens em ordem lógica.
- Se GMASGA for especificado sem GMLOGO, GMASGA sempre terá um efeito. Isso significa que a opção deve ser desativada após o primeiro segmento na mensagem lógica ter sido removido da fila, para poder remover os segmentos restantes na mensagem lógica.

Se esta opção não for especificada, os segmentos de mensagem poderão ser recuperados mesmo quando a mensagem lógica estiver incompleta...

Enquanto o GMCMPM e o GMASGA requerem que todos os segmentos estejam disponíveis antes que qualquer um deles possa ser recuperado, o primeiro retorna a mensagem completa, enquanto o segundo permite que os segmentos sejam recuperados um por um.

Se GMASGA for especificado para uma mensagem de relatório, o gerenciador de fila executa processamento especial. O gerenciador de filas verifica a fila para ver se há pelo menos uma mensagem de relatório para cada um dos segmentos que compõem a mensagem lógica completa. Se houver, a condição GMASGA será satisfeita. No entanto, o gerenciador de filas não verifica o tipo das mensagens de relatório presentes e, portanto, pode haver uma mistura de tipos de relatório nas mensagens de relatório relacionadas aos segmentos da mensagem lógica... Como resultado, o sucesso do GMASGA não implica que o GMCMPM seja bem-sucedido. Se houver

uma combinação de tipos de relatórios presentes para os segmentos de uma mensagem lógica específica, essas mensagens de relatórios devem ser recuperados uma por uma.

GMASGA pode ser especificado com qualquer uma das outras opções GM\* e com qualquer uma das opções MO\*.

**Opção padrão:** se nenhuma das opções descritas for necessária, a seguinte opção poderá ser usada:

#### **GMNONE**

Nenhuma opção especificada.

Este valor pode ser utilizado para indicar que nenhuma outra opção foi especificada; todas as opções assumem seus valores padrão.. GMNONE é definido para auxiliar a documentação do programa; não se pretende que esta opção seja usada com qualquer outra, mas como seu valor é zero, tal uso não pode ser detectado.

O valor inicial do campo *GMOPT* é GMNWT.

#### **GMRE1 (sequência de caracteres de 1 byte)**

Reservado.

Este é um campo reservado O valor inicial desse campo é um caractere em branco. Esse campo será ignorado se *GMVER* for menor que GMVER2.

#### **GMRL (número inteiro assinado de 10 dígitos)**

Comprimento dos dados da mensagem retornados (bytes).

Esse é um campo de saída configurado pelo gerenciador de filas para o comprimento em bytes dos dados da mensagem retornados pela chamada MQGET no parâmetro **BUFFER** . Se o gerenciador de filas não suportar esse recurso, *GMRL* será configurado para o valor RLUNDF.

Quando as mensagens são convertidas entre codificações ou conjuntos de caracteres, os dados da mensagem às vezes podem mudar de tamanho. No retorno da chamada MQGET:

- Se *GMRL* não for RLUNDF, o número de bytes de dados da mensagem retornados será fornecido por *GMRL*
- Se *GMRL* tiver o valor RLUNDF, o número de bytes de dados da mensagem retornados geralmente será fornecido pelo menor de *BUFLen* e *DATLen*, mas poderá ser menor que isso se a chamada MQGET for concluída com o código de razão RC2079 . Se isso acontecer, os bytes insignificantes no parâmetro **BUFFER** serão configurados como nulos

O seguinte valor especial é definido:

#### **RLUNDF**

Comprimento de dados retornados não definido.

O valor inicial desse campo é RLUNDF.. Esse campo será ignorado se *GMVER* for menor que GMVER3..

#### **GMRQN (sequência de caracteres de 48 bytes).**

Nome da fila de destino resolvido.

Este é um campo de saída configurado pelo gerenciador de filas para o nome local da fila da qual a mensagem foi recuperada, conforme definido para o gerenciador de filas locais. Isso é diferente do nome usado para abrir a fila se:

- Uma fila de alias foi aberta (nesse caso, o nome da fila local para a qual o alias resolvido é retornado) ou
- Uma fila modelo foi aberta (nesse caso, o nome da fila local dinâmica é retornado).

O comprimento deste campo é fornecido por LNQN (LNQN). O valor inicial desse campo é 48 caracteres em branco.

#### **GMRS2 (sequência de caracteres de 1 byte)**

Reservado.



Este é um campo reservado O valor inicial desse campo é um caractere em branco. Esse campo será ignorado se *GMVER* for menor que *GMVER4*

#### **GMSEG (sequência de caracteres de 1 byte).**

Sinalizador indicando se segmentação adicional é permitida para a mensagem recuperada.

Ele possui um dos seguintes valores:

##### **SEGIHB**

Segmentação não permitida.

##### **SEGALW**

Segmentação permitida.

Esse é um campo de saída. O valor inicial desse campo é o SEGIHB. Esse campo será ignorado se *GMVER* for menor que *GMVER2*.

#### **GMSG1 (número inteiro assinado de 10 dígitos)**

Sinal.

Este é um campo reservado; seu valor não é significativo. O valor inicial desse campo é 0.

#### **GMSG2 (número inteiro assinado de 10 dígitos)**

Identificador de sinal.

Este é um campo reservado; seu valor não é significativo.

#### **GMSID (sequência de caracteres de 4 bytes)**

Identificador de estruturação.

O valor deve ser:.

##### **GMSIDV**

Identificador para a estrutura de opções get-message.

Esse campo é sempre um campo de entrada O valor inicial deste campo é GMSIDV.

#### **GMSST (cadeia de caracteres de 1 byte)**

Sinalizador indicando se a mensagem recuperada é um segmento de uma mensagem lógica.

Ele possui um dos seguintes valores:

##### **SSNSEG**

A mensagem não é um segmento.

##### **SSSEG**

A mensagem é um segmento, mas não é o último segmento da mensagem lógica.

##### **SSLSEG**

A mensagem é o último segmento da mensagem lógica.

Este também é o valor retornado se a mensagem lógica consistir em apenas um segmento.

Este campo é um campo de saída O valor inicial deste campo é SSNSEG.. Esse campo será ignorado se *GMVER* for menor que *GMVER2*.

#### **GMTOK (cadeia de bits de 16 bytes)**

Token de mensagem

Este é um campo reservado; seu valor não é significativo. O seguinte valor especial é definido:

##### **MTKNON**

Nenhum token de mensagem..

O valor é binário zero para o comprimento do campo.

O comprimento deste campo é fornecido por LNMTOK O valor inicial deste campo é MTKNON. Esse campo será ignorado se *GMVER* for menor que *GMVER3*..

### **GMVER (número inteiro assinado de 10 dígitos).**

Número de versão da estrutura.

O valor deve ser um dos seguintes:

#### **GMVER1**

Version-1 estrutura de opções get-message.

#### **GMVER2**

Version-2 -estrutura de opções get-message.

#### **GMVER3**

Version-3 estrutura de opções get-message.

#### **GMVER4**

Version-4 estrutura de opções get-message.

Os campos que existem apenas nas versões mais recentes da estrutura são identificados como tal nas descrições dos campos A constante a seguir especifica o número da versão atual:

#### **GMVERC**

A versão atual da estrutura de opções get-message

Esse campo é sempre um campo de entrada O valor inicial desse campo é GMVER1.

### **GMVER (número inteiro assinado de 10 dígitos).**

Número de versão da estrutura.

O valor deve ser um dos seguintes:

#### **GMVER1**

Version-1 estrutura de opções get-message.

#### **GMVER2**

Version-2 -estrutura de opções get-message.

#### **GMVER3**

Version-3 estrutura de opções get-message.

#### **GMVER4**

Version-4 estrutura de opções get-message.

Os campos que existem apenas nas versões mais recentes da estrutura são identificados como tal nas descrições dos campos A constante a seguir especifica o número da versão atual:

#### **GMVERC**

A versão atual da estrutura de opções get-message

Esse campo é sempre um campo de entrada O valor inicial desse campo é GMVER1.

### **GMWI (número inteiro assinado de 10 dígitos)**

Intervalo de espera..

Este é o tempo aproximado, expresso em milissegundos, que a chamada MQGET espera que uma mensagem adequada chegue (ou seja, uma mensagem que satisfaça os critérios de seleção especificados no parâmetro **MSGDSC** da chamada MQGET; consulte o campo *MDMID* descrito em “MQMD (Descritores de mensagens) em IBM i” na página 1148 para obter mais detalhes). Se nenhuma mensagem adequada tiver chegado após esse tempo ter decorrido, a chamada será concluída com CCFAIL e código de razão RC2033.

O *GMWI* é usado com a opção GMWT Será ignorado se essa opção não for especificada. Se for especificado, *GMWI* deverá ser maior ou igual a zero ou o valor especial a seguir:

#### **WIULIM**

Intervalo de espera ilimitado.

O valor inicial desse campo é 0.

## Valores iniciais

Tabela 704. Campos em MQGMO		
Nome de Campo	Nome da constante	Valor da constante
GMSID	GMSIDV	'GMO↵'
GMVER	GMVER1	1
GMOPT	GMNWT	0
GMWI	Nenhum	0
GMSG1	Nenhum	0
GMSG2	Nenhum	0
GMRQN	Nenhum	Espaços em branco
GMMO	MOMSGI + MOCORI	3
GMGST	GSNIG	' '
GMSST	SSNSEG	' '
GMSEG	SEGIHB	' '
GMRE1	Nenhum	' '
GMTOK	MTKNON	Nulos
GMRL	RLUNDF	-1
GMRS2	Nenhum	' '
GMMH	HMNONE	0

### Notas:

1. O símbolo ↵ representa um único caractere em branco.

## Declaração RPG

```

D*.1.....2.....3.....4.....5.....6.....7..
D*
D* MQGMO Structure
D*
D* Structure identifier
D GMSID          1      4   INZ('GMO ')
D* Structure version number
D GMVER          5      8I 0 INZ(1)
D* Options that control the action ofMQGET
D GMOPT          9      12I 0 INZ(0)
D* Wait interval
D GMWI           13     16I 0 INZ(0)
D* Signal
D GMSG1          17     20I 0 INZ(0)
D* Signal identifier
D GMSG2          21     24I 0 INZ(0)
D* Resolved name of destination queue
D GMRQN          25     72   INZ
D* Options controlling selection criteriaused for MQGET
D GMMO           73     76I 0 INZ(3)
D* Flag indicating whether messageretrieved is in a group
D GMGST          77     77   INZ(' ')
D* Flag indicating whether messageretrieved is a segment of a
D* logicalmessage
D GMSST          78     78   INZ(' ')
D* Flag indicating whether furthersegmentation is allowed for themessage
D* retrieved
D GMSEG          79     79   INZ(' ')
D* Reserved

```

D	GMRE1	80	80	INZ
D*	Message token			
D	GMTOK	81	96	INZ(X'0000000000000000-0000000000000000')
D				
D*	Length of message data returned(bytes)			
D	GMRL	97	100I 0	INZ(-1)
D*	Reserved			
D	GMRS2	101	104I 0	INZ(0)
D*	Message handle			
D	GMMH	105	112I 0	INZ(0)

## IBM i MQIIH (IMS cabeçalho de informações) em IBM i

A estrutura MQIIH descreve as informações que devem estar presentes no início de uma mensagem enviada para a ponte do IMS por IBM MQ for z/OS.

### Visão Geral

**Nome do formato:** FMIMS.

**Conjunto de caracteres e codificação:** Condições especiais se aplicam ao conjunto de caracteres e à codificação usados para a estrutura MQIIH e dados de mensagem do aplicativo:

- Aplicativos que se conectam ao gerenciador de filas que possui a fila de ponte IMS devem fornecer uma estrutura MQIIH que esteja no conjunto de caracteres e na codificação do gerenciador de filas. Isso ocorre porque a conversão de dados da estrutura MQIIH não é executada neste caso
- Os aplicativos que se conectam a outros gerenciadores de filas podem fornecer uma estrutura MQIIH que esteja em qualquer um dos conjuntos de caracteres e codificações suportados; a conversão do MQIIH é executada pelo agente do canal de mensagens de recebimento conectado ao gerenciador de filas que possui a fila de ponte do IMS

**Nota:** Há uma exceção para isso. Se o gerenciador de filas que possui a fila de ponte IMS estiver usando CICS para enfileiramento distribuído, o MQIIH deverá estar no conjunto de caractere e na codificação do gerenciador de filas que possui a fila de ponte IMS .

- Os dados da mensagem do aplicativo após a estrutura MQIIH devem estar no mesmo conjunto de caracteres e codificação que a estrutura MQIIH. Os campos *IICSI* e *IIENC* na estrutura MQIIH não podem ser usados para especificar o conjunto de caractere e a codificação dos dados da mensagem do aplicativo

Uma saída de conversão de dados deve ser fornecida pelo usuário para converter os dados de mensagem do aplicativo se os dados não forem um dos formatos integrados suportados pelo gerenciador de filas

- [“Autenticando passtickets para aplicativos de ponte IMS” na página 1136](#)
- [“Campos” na página 1137](#)
- [“Valores iniciais” na página 1140](#)
- [“Declaração RPG” na página 1140](#)

### Autenticando passtickets para aplicativos de ponte IMS

Agora é possível que os administradores do IBM MQ especifiquem o nome do aplicativo a ser usado para autenticar chamados, para aplicativos de ponte do IMS Para fazer isso, o nome do aplicativo é especificado como um novo atributo PTKTAPPL para a definição de objeto STGCLASS, como uma sequência alfanumérica de 1 a 8 caracteres.

Um valor em branco significa que a autenticação ocorre como com liberações anteriores do IBM MQ, ou seja, nenhum nome de aplicativo flui na solicitação de autenticação e o valor MVSxxxx para ser usado no lugar.

Um valor de 1 a 8 caracteres alfanuméricos deve seguir as regras para nomes do aplicativo passticket conforme descrito nas publicações do RACF .

IBM MQ Administradores e RACF administradores devem concordar com os nomes de aplicativos válidos a serem usados. O administrador do RACF deve criar um perfil na classe PTKTDATA que fornece acesso READ aos IDs de usuário de todos os aplicativos que devem receber acesso. O administrador do IBM MQ deve criar ou alterar as definições STGCLASS necessárias que especificam o nome do aplicativo a ser usado para autenticação de passagem.

Para obter informações relacionadas, consulte a *Referência de Comando do Script (MQSC)*.

## Campos

A estrutura MQIIH contém os campos a seguir; os campos são descritos em **ordem alfabética**:

### **IIAUT (cadeia de caractere de 8 bytes).**

Senha ou passticket do RACF

Isso é opcional; se especificado, ele será usado com o ID do usuário no contexto de segurança MQMD para construir um UTOKEN que é enviado para o IMS para fornecer um contexto de segurança. Se ele não for especificado, o ID do usuário será usado sem verificação. Isso depende da configuração dos comutadores RACF, que podem requerer que um autenticador esteja presente..

Isso será ignorado se o primeiro byte estiver em branco ou nulo. O valor especial a seguir pode ser usado:

#### **IAUNON**

Nenhuma autenticação.

O comprimento deste campo é fornecido por LNAUTH O valor inicial desse campo é IAUNON.

### **IICMT (cadeia de caracteres de 1 byte).**

Modo de confirmação...

Consulte o *OTMA Reference* para obter mais informações sobre os modos de confirmação IMS. O valor deve ser um dos seguintes:

#### **ICMCTS**

Confirmar, em seguida, enviar

Esse modo implica em enfileiramento duplo de tempos de ocupação de região de saída, mas mais curtos. Transações de atalho e conversação não podem ser executadas com esse modo.

#### **ICMSTC**

Enviar e, em seguida, confirmar

O valor inicial desse campo é ICMCTS.

### **IICSI (número inteiro assinado de 10 dígitos)**

Reservado.

Este é um campo reservado; seu valor não é significativo. O valor inicial desse campo é 0.

### **IIENC (número inteiro assinado de 10 dígitos)**

Reservado.

Este é um campo reservado; seu valor não é significativo. O valor inicial desse campo é 0.

### **IIFLG (número inteiro assinado de 10 dígitos)**

Sinalizadores.

O valor deve ser:.

#### **IINONE**

Sem bandeiras.

O valor inicial deste campo é IINONE.

**IIFMT (cadeia de caracteres de 8 bytes)**

O nome do formato IBM MQ de dados que seguem MQIIH

Isso especifica o nome do formato IBM MQ dos dados que seguem a estrutura MQIIH

Na chamada MQPUT ou MQPUT1 , o aplicativo deve configurar esse campo para o valor apropriado aos dados. As regras para codificação desse campo são as mesmas que aquelas para o campo *MDFMT* no MQMD.

O comprimento deste campo é fornecido pelo LNFMT O valor inicial deste campo é FMNONE.

**IILEN (número inteiro assinado de 10 dígitos)**

Comprimento da estrutura do MQIIH

O valor deve ser:.

**IILEN1**

Comprimento da estrutura do cabeçalho de informações do IMS

O valor inicial desse campo é IILEN1.

**IILTO (cadeia de caracteres de 8 bytes)**

Substituição do terminal lógico

Isso é colocado no campo PCB de E/S. É opcional; se não for especificado, o nome do TPIPE será usado. Ele será ignorado se o primeiro byte estiver em branco ou nulo.

O comprimento desse campo é fornecido por LNLTOV O valor inicial deste campo é 8 caracteres em branco.

**IIMMN (cadeia de caracteres de 8 bytes).**

Nome do mapa de serviço de formato de mensagem

Isso é colocado no campo PCB de E/S. Ele é opcional. Na entrada ele representa o MID, na saída ele representa o MOD. Ele será ignorado se o primeiro byte estiver em branco ou nulo.

O comprimento deste campo é fornecido por LNMFMN O valor inicial deste campo é 8 caracteres em branco.

**IIRFM (cadeia de caractere de 8 bytes).**

IBM MQ nome do formato da mensagem de resposta..

Este é o nome do formato IBM MQ da mensagem de resposta que será enviada em resposta à mensagem atual As regras para codificar isso são as mesmas para o campo *MDFMT* no MQMD.

O comprimento deste campo é fornecido pelo LNFMT O valor inicial deste campo é FMNONE.

**IIRSV (cadeia de caracteres de 1 byte)**

Reservado.

Este é um campo reservado; ele deve estar em branco

**IISEC (cadeia de caracteres de um byte)**

Escopo de segurança

Isso indica o processamento de segurança do IMS necessário Os seguintes valores são definidos:

**ISSCHK**

Verifique o escopo de segurança

Um ACEE é construído na região de controle, mas não na região dependente

**ISSFUL**

Escopo de segurança total

Um ACEE em cache é construído na região de controle e um ACEE não em cache é construído na região dependente. Se você usar ISSFUL, deverá assegurar que o ID do usuário para o qual o ACEE é construído tenha acesso aos recursos usados na região dependente.

Se ISSCHK e ISSFUL não forem especificados para este campo, ISSCHK será assumido

O valor inicial deste campo é ISSCHK.

#### **IISID (cadeia de caracteres de 4 bytes)**

Identificador de estruturação.

O valor deve ser:.

#### **IISIDV**

Identificador para a estrutura do cabeçalho de informações IMS .

O valor inicial desse campo é IISIDV.

#### **IITID (sequência de bits de 16 bytes).**

Identificador da instância de transação

Esse campo é usado por mensagens de saída do IMS , portanto, é ignorado na primeira entrada Se *IITST* for configurado como ITSIC, ele deverá ser fornecido na próxima entrada e em todas as entradas subsequentes, para permitir que o IMS correlacione as mensagens com a conversa correta O valor especial a seguir pode ser usado:

#### **ITINON**

Nenhum ID de instância de transação.

O comprimento deste campo é fornecido por LNTIID. O valor inicial desse campo é ITINON.

#### **IITST (cadeia de caracteres de 1 byte).**

Estado da Transação

Indica o estado da conversa IMS . Isso é ignorado na primeira entrada porque não existe nenhuma conversa. Em entradas subsequentes, isso indica se uma conversa está ativa ou não; Na saída, ele é configurado por IMS O valor deve ser um dos seguintes:

#### **ITSIC**

Em conversa.

#### **ITSNIC**

Não na conversa.

#### **ITSARC**

Retornar dados de estado da transação no formato arquitetado.

Esse valor é usado somente com o comando IMS /DISPLAY TRAN .. Isso faz com que os dados de estado de transação sejam retornados no formato arquitetado IMS em vez de no formato de caractere Consulte [Gravando IMS programas de transações por IBM MQ](#) para obter detalhes adicionais.

O valor inicial deste campo é ITSNIC.

#### **IIVER (número inteiro assinado de 10 dígitos)**

Número de versão da estrutura.

O valor deve ser:.

#### **IIVER1**

Número da versão para a estrutura do cabeçalho de informações do IMS

A constante a seguir especifica o número da versão atual:

#### **IIVERC**

Versão atual da estrutura do cabeçalho de informações IMS .

O valor inicial desse campo é IIVER1.

## Valores iniciais

Tabela 705. Campos em MQIIH		
Nome de Campo	Nome da constante	Valor da constante
IISID	IISIDV	'IIH~'
IIVER	IIVER1	1
IILEN	IILEN1	84
IIENC	Nenhum	0
IICSI	Nenhum	0
IIFMT	FMNONE	Espaços em branco
IIFLG	IINONE	0
IILTO	Nenhum	Espaços em branco
IIMMN	Nenhum	Espaços em branco
IIRFM	FMNONE	Espaços em branco
IIAUT	IAUNON	Espaços em branco
IITID	ITINON	Nulos
IITST	ITSNIC	' '
IICMT	ICMCTS	'0'
IISEC	ISSCHK	'C'
IIRSV	Nenhum	' '

### Notas:

1. O símbolo ~ representa um único caractere em branco.

## Declaração RPG

```

D*.1.....2.....3.....4.....5.....6.....7..
D*
D* MQIIH Structure
D*
D* Structure identifier
D IISID          1      4  INZ('IIH ')
D* Structure version number
D IIVER          5      8I 0 INZ(1)
D* Length of MQIIH structure
D IILEN          9     12I 0 INZ(84)
D* Reserved
D IIENC          13     16I 0 INZ(0)
D* Reserved
D IICSI          17     20I 0 INZ(0)
D* MQ format name of data that followsMQIIH
D IIFMT          21     28  INZ('      ')
D* Flags
D IIFLG          29     32I 0 INZ(0)
D* Logical terminal override
D IILTO          33     40  INZ
D* Message format services map name
D IIMMN          41     48  INZ
D* MQ format name of reply message
D IIRFM          49     56  INZ('      ')
D* RACF password or passticket
D IIAUT          57     64  INZ('      ')
D* Transaction instance identifier
D IITID          65     80  INZ(X'00000000000000-
D                                0000000000000000')

```



D*	Transaction state			
D	IITST	81	81	INZ(' ')
D*	Commit mode			
D	IICMT	82	82	INZ('0')
D*	Security scope			
D	IISEC	83	83	INZ('C')
D*	Reserved			
D	IIRSV	84	84	INZ

## IBM i MQIMPO (Consultar opções de propriedade de mensagem) no IBM i

A estrutura MQIMPO permite que aplicativos especifiquem opções que controlam como as propriedades de mensagens são consultadas.

### Visão Geral

**Propósito:** a estrutura é um parâmetro de entrada na chamada MQINQMP.

**Conjunto de caracteres e codificação:** os dados em MQIMPO devem estar no conjunto de caracteres do aplicativo e codificação do aplicativo (ENNAT).

- [“Campos” na página 1141](#)
- [“Valores iniciais” na página 1147](#)
- [“Declaração RPG” na página 1147](#)

### Campos

A estrutura MQIMPO contém os campos a seguir; os campos são descritos em **ordem alfabética**:

#### IPOPT (número inteiro assinado de 10 dígitos)

As opções a seguir controlam a ação de MQINQMP. É possível especificar uma ou mais dessas opções. Para especificar mais de uma opção, inclua os valores juntos (não inclua a mesma constante mais de uma vez) ou combine os valores usando a operação OR bit a bit (se a linguagem de programação suportar operações de bit). As combinações de opções inválidas são anotadas; todas as outras combinações são válidas.

**Opções de dados de valor:** As opções a seguir estão relacionadas ao processamento dos dados de valor quando a propriedade é recuperada da mensagem.

#### IPCVAL

Essa opção solicita que o valor da propriedade seja convertido para estar em conformidade com os valores *IPREQCSI* e *IPREQENC* especificados antes da chamada MQINQMP retornar o valor da propriedade na área *Value*.

- Se a conversão for bem-sucedida, os campos *IPRETCSI* e *IPRETENC* serão configurados como *IPREQCSI* e *IPREQENC* no retorno da chamada MQINQMP.
- Se a conversão falhar, mas a chamada MQINQMP for concluída sem erro, o valor da propriedade será retornado sem conversão.

Se a propriedade for uma sequência, os campos *IPRETCSI* e *IPRETENC* serão configurados para o conjunto de caracteres e a codificação da sequência não convertida.

Neste caso, o código de conclusão é CCWARN, com o código de razão RC2466. O cursor da propriedade é avançado para a propriedade retornada.

Se o valor da propriedade se expandir durante a conversão e exceder o tamanho do parâmetro **Value**, o valor será retornado não convertido, com o código de conclusão CCFAIL; o código de razão é configurado como RC2469.

O parâmetro **DataLength** da chamada MQINQMP retorna o comprimento para o qual o valor da propriedade teria sido convertido, para permitir que o aplicativo determine o tamanho do

buffer necessário para acomodar o valor da propriedade convertido. O cursor da propriedade está inalterado

Essa opção também solicita que:

- Se o nome da propriedade contiver um curinga e
- O campo *IPRETNAMECHRP* é inicializado com um endereço ou deslocamento para o nome retornado,

então o nome retornado será convertido para estar em conformidade com os valores *IPREQCSI* e *IPREQENC*.

- Se a conversão for bem-sucedida, o campo *VSCCSID* do *IPRETNAMECHRP* e a codificação do nome retornado serão configurados para o valor de entrada de *IPREQCSI* e *IPREQENC*.
- Se a conversão falhar, mas a chamada *MQINQMP* for concluída sem erro ou aviso, o nome retornado não será convertido. O código de conclusão é *CCWARN* neste caso, com código de razão *RC2492*.

O cursor da propriedade é avançado para a propriedade retornada *RC2466* será retornado se o valor e o nome não forem convertidos.

Se o nome retornado se expandir durante a conversão e exceder o tamanho do campo *VSBuFSIZE* do *RequestedName*, a sequência retornada será deixada sem conversão, com o código de conclusão *CCFAIL* e o código de razão será configurado como *RC2465*.

O campo *VSLength* da estrutura *MQCHARV* retorna o comprimento para o qual o valor da propriedade teria sido convertido, para permitir que o aplicativo determine o tamanho do buffer necessário para acomodar o valor da propriedade convertido. O cursor da propriedade está inalterado

#### **IPCTYP**

Esta opção solicita que o valor da propriedade seja convertido de seu tipo de dados atual para o tipo de dados especificado no parâmetro **Type** da chamada *MQINQMP*

- Se a conversão for bem-sucedida, o parâmetro **Type** permanecerá inalterado no retorno da chamada *MQINQMP*
- Se a conversão falhar, mas a chamada *MQINQMP* for concluída sem erro, a chamada falhará com o motivo *RC2470*. O cursor da propriedade está inalterado

Se a conversão do tipo de dados fizer com que o valor seja expandido durante a conversão e o valor convertido exceder o tamanho do parâmetro **Value**, o valor será retornado não convertido, com o código de conclusão *CCFAIL* e o código de razão configurado como *RC2469*.

O parâmetro **DataLength** da chamada *MQINQMP* retorna o comprimento para o qual o valor da propriedade teria sido convertido, para permitir que o aplicativo determine o tamanho do buffer necessário para acomodar o valor da propriedade convertido. O cursor da propriedade está inalterado

Se o valor do parâmetro **Type** da chamada *MQINQMP* não for válido, a chamada falhará com o motivo *RC2473*.

Se a conversão de tipo de dados solicitada não for suportada a chamada falha com a razão *RC2470*. As seguintes conversões de tipo de dados são suportadas:

<i>Tabela 706. Conversões de tipo de dados suportadas</i>	
<b>Tipo de Dados da Propriedade</b>	<b>Tipos de dados de destino suportados</b>
TYPBOL	TYPSTR, TYPI8, TYPI16, TYPI32 TYPI64
TYPBST	TYPSTR
TYPI8	TYPSTR, TYPI16, TYPI32, TYPI64
TYPI16	TYPSTR, TYPI32, TYPI64

Tabela 706. Conversões de tipo de dados suportadas (continuação)

Tipo de Dados da Propriedade	Tipos de dados de destino suportados
TYPI32	TYPSTR, TYPI64 .
TYPI64	TYPSTR
TYPF32	TYPSTR, TYPF64
TYPF64	TYPSTR
TYPSTR	TYPBOL, TYPI8, TYPI16, TYPI32, TYPI64, TYPF32, TYPF64
TYPNUL	Nenhum

As regras gerais que regem as conversões suportadas são as seguintes:

- Valores de propriedade numérica podem ser convertidos de um tipo de dados para outro, desde que nenhum dado seja perdido durante a conversão.

Por exemplo, o valor de uma propriedade com tipo de dados TYPI32 pode ser convertido em um valor com tipo de dados TYPI64, mas não pode ser convertido em um valor com tipo de dados TYPI16.

- Um valor de propriedade de qualquer tipo de dados pode ser convertido em uma sequência.
- Um valor de propriedade de sequência pode ser convertido para qualquer outro tipo de dados, desde que a sequência seja formatada corretamente para a conversão. Se um aplicativo tentar converter um valor de propriedade de cadeia que não esteja formatado corretamente, IBM MQ retornará o código de razão RC2472.
- Se um aplicativo tentar uma conversão que não seja suportada, o IBM MQ retornará o código de razão RC2470

As regras específicas para converter um valor de propriedade de um tipo de dados para outro são as seguintes:

- Ao converter um valor da propriedade TYPBOL em uma cadeia, o valor TRUE é convertido na cadeia "TRUE" e o valor false é convertido na cadeia "FALSE".
- Ao converter um valor da propriedade TYPBOL em um tipo de dados numérico, o valor TRUE é convertido em um e o valor FALSE é convertido em zero.
- Ao converter um valor de propriedade de cadeia para um valor TYPBOL, a cadeia "TRUE", ou "1", é convertida para TRUE e a cadeia "FALSE", ou "0", é convertida para FALSE.

Observe que os termos "TRUE" e "FALSE" não fazem distinção entre maiúsculas e minúsculas.

Nenhuma outra sequência pode ser convertida; IBM MQ retorna o código de razão RC2472.

- Ao converter um valor de propriedade de cadeia para um valor com o tipo de dados TYPI8, TYPI16, TYPI32 ou TYPI64, a cadeia deve ter o seguinte formato:

```
[blanks][sign]digits
```

Os significados dos componentes da sequência são os seguintes:

**blanks**

Caracteres em branco iniciais opcionais

**sign**

Um caractere de sinal de mais (+) ou de sinal de menos (-) opcional.

**digits**

Uma sequência contígua de caracteres de dígito (0-9). Pelo menos um caractere de dígito deve estar presente.

Após a sequência de caracteres de dígito, a sequência pode conter outros caracteres que não são caracteres de dígito, mas a conversão para assim que o primeiro desses caracteres for atingido. A sequência é assumida para representar um número inteiro decimal.

IBM MQ retorna o código de razão RC2472 se a sequência não estiver formatada corretamente.

- Ao converter um valor da propriedade de cadeia para um valor com o tipo de dado TYPF32 ou TYPF64, a cadeia deve ter o seguinte formato:

```
[blanks][sign]digits[.digits][e_char[e_sign]e_digits]
```

Os significados dos componentes da sequência são os seguintes:

**blanks**

Caracteres em branco iniciais opcionais

**sign**

Um caractere de sinal de mais (+) ou de sinal de menos (-) opcional.

**digits**

Uma sequência contígua de caracteres de dígito (0-9). Pelo menos um caractere de dígito deve estar presente.

**e\_char**

Um caractere de expoente, que é "E" ou "e".

**e\_sign**

Um caractere de sinal de mais (+) ou sinal de menos (-) opcional para o expoente.

**e\_digits**

Uma sequência contígua de caracteres de dígito (0-9) para o expoente. Pelo menos um caractere de dígito deve estar presente se a sequência contiver um caractere expoente.

Após a sequência de caracteres de dígito ou os caracteres opcionais que representam um expoente, a sequência pode conter outros caracteres que não são caracteres de dígito, mas a conversão para assim que o primeiro desses caracteres for atingido. Supõe-se que a sequência represente um número de vírgula flutuante decimal com um expoente que é uma potência de 10.

IBM MQ retorna o código de razão RC2472 se a sequência não estiver formatada corretamente.

- Ao converter um valor de propriedade numérica em uma sequência, o valor é convertido para a representação de sequência do valor como um número decimal, não a sequência que contém o caractere ASCII para esse valor... Por exemplo, o número inteiro 65 é convertido para a sequência "65", não a sequência "A".
- Ao converter um valor de propriedade de sequência de bytes em uma sequência, cada byte é convertido para os dois caracteres hexadecimais que representam o byte... Por exemplo, a matriz de bytes {0xF1, 0x12, 0x00, 0xFF} é convertida na sequência "F11200FF".

**IPQLEN**

Consulte o tipo e comprimento do valor da propriedade. O comprimento é retornado no parâmetro **DataLength** da chamada MQINQMP.. O valor da propriedade não é retornado.

Se um buffer *ReturnedName* for especificado, o campo *VSLength* da estrutura MQCHARV será preenchido com o comprimento do nome da propriedade.. O nome da propriedade não é retornado..

**Opções de iteração:** As opções a seguir estão relacionadas à iteração sobre propriedades, usando um nome com um caractere curinga

**IPINQF**

Consulte a primeira propriedade correspondente ao nome especificado. Após essa chamada, um cursor é estabelecido na propriedade retornada.

Esse é o valor-padrão.

A opção IPINQC pode ser usada subsequentemente com uma chamada MQINQMP, se necessário, para consultar a mesma propriedade novamente.

Observe que há apenas um cursor de propriedade; portanto, se o nome da propriedade, especificado na chamada MQINQMP, alterar o cursor será reconfigurado

Esta opção não é válida com uma das seguintes opções:

IPINQN

IPINQC

### **IPINQN**

Consulta sobre a próxima propriedade que corresponde ao nome especificado, continuando a procura a partir do cursor de propriedade O cursor é avançado para a propriedade retornada.

Se esta for a primeira chamada MQINQMP para o nome especificado, então a primeira propriedade que corresponde ao nome especificado será retornada

A opção IPINQC pode ser usada subsequentemente com uma chamada MQINQMP, se necessário, para consultar a mesma propriedade novamente.

Se a propriedade sob o cursor tiver sido excluída, MQINQMP retornará a próxima propriedade correspondente após aquela que foi excluída

Se for incluída uma propriedade que corresponda ao curinga, enquanto uma iteração estiver em andamento, a propriedade poderá ou não ser retornada durante a conclusão da iteração A propriedade é retornada quando a iteração é reiniciada usando IPINQF.

Uma propriedade correspondente ao curinga que foi excluído, enquanto a iteração estava em andamento, não é retornada após a sua exclusão

Esta opção não é válida com uma das seguintes opções:

IPINQF

IPINQC

### **IPINQC**

Recuperar o valor da propriedade apontada pelo cursor da propriedade. A propriedade apontada pelo cursor da propriedade é aquela que foi consultada pela última vez, usando a opção IPINQF ou IPINQN..

O cursor de propriedade é reconfigurado quando o identificador de mensagens é reutilizado, quando o identificador de mensagens é especificado no campo *MsgHandle* do MQGMO em uma chamada MQGET ou quando o identificador de mensagens é especificado nos campos *OriginalMsgHandle* ou *NewMsgHandle* da estrutura MQPMO em uma chamada MQPUT

Se essa opção for usada quando o cursor de propriedade ainda não tiver sido estabelecido ou se a propriedade apontada pelo cursor de propriedade tiver sido excluída, a chamada falhará com o código de conclusão CCFAIL e a razão RC2471

Esta opção não é válida com uma das seguintes opções:

IPINQF

IPINQN

Se nenhuma das opções descritas anteriormente for necessária, a seguinte opção poderá ser usada:

### **IPNONE**

Use esse valor para indicar que nenhuma outra opção foi especificada. Todas as opções assumem seus valores-padrão.

IPNONE auxilia a documentação do programa; não se pretende que esta opção seja usada com qualquer outra, mas como seu valor é zero, tal uso não pode ser detectado.

Isto é sempre um campo de entrada. O valor inicial deste campo é IPINQF.

### **IPREQCSI (número inteiro assinado de 10 dígitos).**

O conjunto de caracteres em que o valor da propriedade consultada deve ser convertido se o valor for uma cadeia de caracteres. Esse também é o conjunto de caracteres no qual o *ReturnedName* deve ser convertido quando IPCVAL ou IPCTYP for especificado

O valor inicial desse campo é CSAPL.

### **IPREQENC (número inteiro assinado de 10 dígitos)**

Esta é a codificação na qual o valor da propriedade consultada deve ser convertido quando IPCVAL ou IPCTYP for especificado.

O valor inicial deste campo é ENNAT

### **IPRE1 (número inteiro assinado de 10 dígitos)**

Este é um campo reservado O valor inicial desse campo é um caractere em branco.

### **IPRETCSI (número inteiro assinado de 10 dígitos).**

Na saída, esse é o conjunto de caracteres do valor retornado se o parâmetro **Type** da chamada MQINQMP for TYPSTR.

Se a opção IPCVAL for especificada e a conversão for bem-sucedida, o campo *ReturnedCCSID* , no retorno, será o mesmo valor que o valor transmitido.

O valor inicial deste campo é zero.

### **IPRETENC (número inteiro assinado de 10 dígitos)**

Na saída, esta é a codificação do valor retornado.

Se a opção IPCVAL for especificada e a conversão for bem-sucedida, o campo *ReturnedEncoding* , no retorno, será o mesmo valor que o valor transmitido.

O valor inicial deste campo é ENNAT

### **IPRETNAMCHRP (número inteiro assinado de 10 dígitos)**

O nome real da propriedade consultada..

Na entrada, um buffer de sequência pode ser transmitido usando o campo *VSPtr* ou *VSOffset* da estrutura MQCHARV O comprimento do buffer de sequência é especificado usando o campo *VSBuFSIZE* da estrutura MQCHARV.

No retorno da chamada MQINQMP, o buffer de sequência é concluído com o nome da propriedade que foi consultada, contanto que o buffer de sequência tenha sido longo o suficiente para conter totalmente o nome. O campo *VSLength* da estrutura MQCHARV é preenchido com o comprimento do nome da propriedade.. O campo *VSCCSID* da estrutura MQCHARV é preenchido para indicar o conjunto de caracteres do nome retornado, se a conversão do nome falhou.

Este é um campo de entrada/saída O valor inicial desse campo é MQCHARV\_DEFAULT.

### **IPSID (número inteiro assinado de 10 dígitos)**

Este é o identificador de estrutura. O valor deve ser:.

#### **IPSIDV**

Identificador para a estrutura de opções de propriedade de mensagem de consulta.

Isto é sempre um campo de entrada. O valor inicial desse campo é IPSIDV.

## **IPITYP (número inteiro assinado de 10 dígitos)**

Uma representação em cadeia do tipo de dados da propriedade

Se a propriedade foi especificada em um cabeçalho MQRFH2 e o atributo MQRFH2 dt não for reconhecido, esse campo poderá ser usado para determinar o tipo de dado da propriedade.

*TypeString* é retornado no conjunto de caracteres codificados 1208 (UTF-8) e é os primeiros oito bytes do valor do atributo dt da propriedade que falhou ao ser reconhecido

Este é sempre um campo de saída O valor inicial deste campo é a cadeia nula na linguagem de programação C e 8 caracteres em branco em outras linguagens de programação.

## **IPVER (número inteiro assinado de 10 dígitos)**

Este é o número da versão da estrutura O valor deve ser:.

### **IPVER1**

Número da versão para a estrutura de opções de propriedades de mensagem de consulta

A constante a seguir especifica o número da versão atual:

### **IPVERC**

Versão atual da estrutura de opções de propriedades de mensagem de consulta

Isto é sempre um campo de entrada. O valor inicial desse campo é IPVER1.

## **Valores iniciais**

<i>Tabela 707. Campos em MQIPMO</i>		
<b>Nome de Campo</b>	<b>Nome da constante</b>	<b>Valor da constante</b>
<i>IPSID</i>	IPSIDV	'IMPO'
<i>IPVER</i>	IPVER1	1
<i>IPOPT</i>	IPINQF	
<i>IPREQENC</i>	ENNAT	
<i>IPREQCSI</i>	CSAPL	
<i>IPRETENC</i>	ENNAT	
<i>IPRETCSI</i>	0	
<i>IPRE1</i>	0	
<i>IPRETNAMCHRP</i>		
<i>IPITYP</i>		em branco

## **Declaração RPG**

```
D* MQIMPO Structure
D*
D*
D* Structure identifier
D IPSID      1  4  INZ('IMPO')
D*
D* Structure version number
D IPVER      5  8I 0  INZ(1)
D*
** Options that control the action of
D* MQINQMP
D IPOPT      9  12I 0  INZ(0)
D*
D* Requested encoding of Value
```

```

D IPREQENC      13   16I 0 INZ(273)
D*
** Requested character set identifier
D* of Value
D IPREQCSI      17   20I 0 INZ(-3)
D*
D* Returned encoding of Value
D IPRETENC      21   24I 0 INZ(273)
D*
** Returned character set identifier of
D* Value
D IPRETCSI      25   28I 0 INZ(0)
D*
D* Reserved
D IPRE1         29   32I 0 INZ(0)
D*
D* Returned property name
D* Address of variable length string
D IPRETAMCHRP   33   48* INZ(*NULL)
D* Offset of variable length string
D IPRETAMCHRO   49   52I 0 INZ(0)
D* Size of buffer
D IPRETAMVSBS   53   56I 0 INZ(-1)
D* Length of variable length string
D IPRETAMCHRL   57   60I 0 INZ(0)
D* CCSID of variable length string
D IPRETAMCHRC   61   64I 0 INZ(-3)
D*
D* Property data type as a string
D IPTYP        65   72 INZ

```

## IBM i MQMD (Descritores de mensagens) em IBM i

### Visão Geral

**Propósito:** a estrutura MQMD contém as informações de controle que acompanham os dados do aplicativo quando uma mensagem viaja entre os aplicativos de envio e recebimento. A estrutura é um parâmetro de entrada / saída nas chamadas MQGET, MQPUT e MQPUT1 ..

**Versão:** A versão atual do MQMD é MDVER2. Os campos que existem apenas nas versões mais recentes da estrutura são identificados como tal nas descrições a seguir:

O arquivo COPY fornecido contém a versão mais recente do MQMD que é suportada pelo ambiente, mas com o valor inicial do campo MDVER configurado como MDVER1... Para usar campos que não estão presentes na estrutura version-1 , o aplicativo deve configurar o campo MDVER para o número da versão necessária.

Uma declaração para a estrutura version-1 está disponível com o nome MQMD1.

**Conjunto de caracteres e codificação:** os dados no MQMD devem estar no conjunto de caracteres fornecido pelo atributo do gerenciador de filas do **CodedCharSetId** e a codificação do gerenciador de filas local fornecida pelo ENNAT... No entanto, se o aplicativo estiver em execução como um IBM MQ MQI client, a estrutura deverá estar no conjunto de caracteres e na codificação do cliente.

Se os gerenciadores de filas de envio e de recebimento usarem conjuntos de caracteres ou codificações diferentes, os dados no MQMD serão convertidos automaticamente. Não é necessário que o aplicativo converta o MQMD.

- [“Usando diferentes versões de MQMD” na página 1149](#)
- [“Contexto da mensagem” na página 1149](#)
- [“Expiração da mensagem” na página 1149](#)
- [“Campos” na página 1150](#)
- [“Valores iniciais” na página 1192](#)
- [“Declaração RPG” na página 1193](#)



## Usando diferentes versões de MQMD

Um MQMD version-2 geralmente é equivalente a usar um MQMD version-1 e prefixar os dados da mensagem com uma estrutura MQMDE. No entanto, se todos os campos na estrutura MQMDE tiverem seus valores padrão, MQMDE poderá ser omitido. Um MQMD version-1 mais MQMDE são usados conforme descrito posteriormente nesta seção

- Nas chamadas MQPUT e MQPUT1 , se o aplicativo fornecer um MQMD version-1 , o aplicativo poderá opcionalmente prefixar os dados da mensagem com um MQMDE, configurando o campo MDFMT em MQMD para FMMDE para indicar que um MQMDE está presente. Se o aplicativo não fornecer um MQMDE, o gerenciador de filas assume valores padrão para os campos no MQMDE.

**Nota:** Vários dos campos que existem no MQMD version-2 mas não no MQMD version-1 são campos de entrada / saída nas chamadas MQPUT e MQPUT1 . No entanto, o gerenciador de filas não retorna nenhum valor nos campos equivalentes no MQMDE na saída das chamadas MQPUT e MQPUT1 ; se o aplicativo requerer esses valores de saída, ele deverá usar um MQMD version-2 .

- Na chamada MQGET, se o aplicativo fornecer um MQMD version-1 , o gerenciador de filas prefixa a mensagem retornada com um MQMDE, mas somente se um ou mais dos campos no MQMDE tiver um valor não padrão. O campo MDFMT em MQMD terá o valor FMMDE para indicar que um MQMDE está presente.

Os valores padrão que o gerenciador de fila usou para os campos no MQMDE são iguais aos valores iniciais desses campos, mostrados em [Tabela 709 na página 1192](#).

Quando uma mensagem está em uma fila de transmissão, alguns dos campos no MQMD são configurados para valores específicos; consulte [“MQXQH \(Transmission-queue header\) em IBM i” na página 1290](#) para obter detalhes

## Contexto da mensagem

Determinados campos no MQMD contêm o contexto de mensagem Tipicamente:

- *Contexto de identidade* relacionado ao aplicativo que originalmente colocou a mensagem
- *Contexto de origem* relacionado ao aplicativo que colocou a mensagem mais recentemente
- *Contexto do usuário* está relacionado ao aplicativo que originalmente colocou a mensagem.

Esses dois aplicativos podem ser o mesmo aplicativo, mas eles também podem ser aplicativos diferentes (por exemplo, quando uma mensagem é encaminhada de um aplicativo para outro)

Embora o contexto de identidade e origem geralmente tenha os significados descritos anteriormente, o conteúdo dos dois tipos de campos de contexto no MQMD realmente depende das opções PM\* que são especificadas quando a mensagem é colocada. Como resultado, o contexto de identidade não está necessariamente relacionado ao aplicativo que originalmente colocou a mensagem e o contexto de origem não está necessariamente relacionado ao aplicativo que mais recentemente colocou a mensagem-ele depende do design do conjunto de aplicativos.

Há uma classe de aplicativo que nunca altera o contexto da mensagem, ou seja, o agente do canal de mensagens (MCA).. MCAs que recebem mensagens de gerenciadores de filas remotas usam a opção de contexto PMSETA na chamada MQPUT ou MQPUT1 . Isso permite que o MCA de recebimento preserve exatamente o contexto da mensagem que viajou com a mensagem do MCA de envio. No entanto, o resultado é que o contexto de origem não está relacionado ao aplicativo que mais recentemente colocou a mensagem (o MCA de recebimento), mas, em vez disso, está relacionado a um aplicativo anterior que colocou a mensagem (possivelmente o próprio aplicativo de origem).

Para obter informações adicionais, consulte [Contexto da mensagem](#)

## Expiração da mensagem

Mensagens que expiraram em uma fila carregada (uma fila que foi aberta) são removidas automaticamente da fila dentro de um período de tempo razoável após sua expiração. Alguns outros novos recursos desta liberação do IBM MQ podem fazer com que as filas carregadas sejam varridas com

menos frequência do que na versão anterior do produto, no entanto, as mensagens expiradas nas filas carregadas são sempre removidas dentro de um período razoável de sua expiração

## Campos

A estrutura MQMD contém os campos a seguir; os campos são descritos em ordem alfabética:



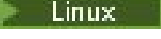

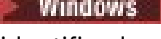
### MDACC (sequência de bits de 32 bytes).

Token de Contabilidade.

Isso faz parte do *contexto de identidade* da mensagem. Para obter mais informações sobre o contexto de mensagem, consulte [Contexto da mensagem](#) e [Informações de contexto de controle](#).

*MDACC* permite que um aplicativo faça com que o trabalho feito como resultado da mensagem seja adequadamente cobrado. O gerenciador de filas trata estas informações como uma sequência de bits e não verifica seu conteúdo.

Quando o gerenciador de filas gera essas informações, ele é configurado como a seguir:

- O primeiro byte do campo é configurado para o comprimento das informações de contabilidade presentes nos bytes a seguir; esse comprimento está no intervalo de zero a 30 e é armazenado no primeiro byte como um número inteiro binário.
- O segundo e os bytes subsequentes (conforme especificado pelo campo de comprimento) são configurados para as informações de contabilidade apropriadas para o ambiente.
  -  **z/OS** No z/OS, as informações de contabilidade são configuradas como:
    - Para o lote z/OS, as informações de contabilidade do cartão JES JOB ou de uma instrução JES ACCT no cartão EXEC (separadores de vírgula são alterados para X'FF '). Essas informações são truncadas, se necessário, para 31 bytes
    - Para o TSO, o número da conta do usuário
    - Para CICS, o identificador da unidade de trabalho LU 6.2 (UEPUOWDS) (26 bytes).
    - Para IMS, o nome do PSB de 8 caracteres concatenado com o token de recuperação de 16 caracteres IMS
  -  **IBM i** No IBM i, as informações de contabilidade são configuradas para o código de contabilidade para a tarefa
  -  **Linux**  **AIX** No AIX and Linux, as informações de contabilidade são configuradas para o identificador de usuário numérico, em caracteres ASCII.
  -  **Windows** No Windows, as informações de contabilidade são configuradas para um identificador de segurança (SID) Windows NT em um formato compactado. O SID identifica exclusivamente o identificador de usuário armazenado no campo *MDUID*. Quando o SID é armazenado no campo *MDACC*, a Autoridade do Identificador de 6 bytes (localizada no terceiro e nos bytes subsequentes do SID) é omitida. Por exemplo, se o SID Windows NT tiver 28 bytes de comprimento, 22 bytes de informações do SID serão armazenados no campo *MDACC*.
- O último byte é configurado para o tipo de token de contabilidade, um dos seguintes valores:

#### **ATTCIC**

CICS identificador de LUOW

#### **ATTDOS**

Token de contabilidade padrão do PC DOS.

#### **ATTWNT**

Identificador de segurança Windows .

#### **ATT400**

Token de contabilidade IBM i .

**ATTUNX**

AIX and Linux identificador numérico..

**ATTUSR**

Token de contabilidade definido pelo usuário..

**ATTUNK**

Tipo de token contábil desconhecido

O tipo de token de contabilidade é configurado para um valor explícito apenas nos seguintes ambientes:

-  AIX
-  IBM i
-  Windows

e para o IBM MQ MQI clients conectado a esses sistemas

Em outros ambientes, o tipo de token contábil é configurado para o valor ATTUNK. Nesses ambientes o campo MDPAT pode ser usado para deduzir o tipo de token de contabilidade recebido.

- Todos os outros bytes são configurados como zero binário.

Para as chamadas MQPUT e MQPUT1, este é um campo de entrada / saída se PMSETI ou PMSETA for especificado no parâmetro **PMO**. Se nem PMSETI nem PMSETA forem especificados, este campo será ignorado na entrada e será um campo somente de saída. Para obter mais informações sobre o contexto da mensagem, consulte [Contexto da mensagem](#) e [Informações sobre o contexto de controle](#)

Após a conclusão bem-sucedida de uma chamada MQPUT ou MQPUT1, esse campo contém o MDACC que foi transmitido com a mensagem se ele foi colocado em uma fila. Este será o valor de MDACC que será mantido com a mensagem se ela for retida (consulte a descrição de PMRET em [“MQPMO \(opções Put-message\) em IBM i”](#) na [página 1215](#) para obter mais detalhes sobre as publicações retidas), mas não será usado como o MDACC quando a mensagem for enviada como uma publicação para os assinantes, pois eles fornecem um valor para substituir MDACC em todas as publicações enviadas para eles.. Se a mensagem não tiver nenhum contexto, o campo será totalmente binário zero.

Este é um campo de saída para a chamada MQGET

Esse campo não está sujeito a nenhuma conversão baseada no conjunto de caracteres do gerenciador de filas-o campo é tratado como uma sequência de bits e não como uma sequência de caracteres.

O gerenciador de filas não faz nada com as informações neste campo O aplicativo deve interpretar as informações se ele desejar usar as informações para fins contábeis

O seguinte valor especial pode ser usado para o campo *MDACC* :

**ACNONE**

Nenhum token de contabilidade especificado.

O valor é binário zero para o comprimento do campo.

O comprimento deste campo é fornecido por LMACCT O valor inicial deste campo é ACNONE.

**MDAID (sequência de caracteres de 32 bytes).**

Dados de aplicativos relacionados à identidade.

Isso faz parte do *contexto de identidade* da mensagem Para obter mais informações sobre o contexto de mensagem, consulte [Contexto da mensagem](#) e [Informações de contexto de controle](#).

MDAID são informações definidas pelo conjunto de aplicativos e podem ser usadas para fornecer informações adicionais sobre a mensagem ou seu originador. O gerenciador de filas trata essas informações como dados de caractere, mas não define o formato deles. Quando o gerenciador de filas gera essas informações, ele fica totalmente em branco

Para as chamadas MQPUT e MQPUT1 , este é um campo de entrada / saída se PMSETI ou PMSETA for especificado no parâmetro **PMO** . Se um caractere nulo estiver presente, o caractere nulo e quaisquer caracteres seguintes serão convertidos em espaços em branco pelo gerenciador de filas. Se nem PMSETI nem PMSETA forem especificados, este campo será ignorado na entrada e será um campo somente de saída Para obter mais informações sobre o contexto da mensagem, consulte [Contexto da mensagem](#) e [Informações sobre o contexto de controle](#)

Após a conclusão bem-sucedida de uma chamada MQPUT ou MQPUT1 , esse campo contém o MDAID que foi transmitido com a mensagem se ele foi colocado em uma fila. Esse será o valor de MDAID que será mantido com a mensagem se ela for retida (consulte a descrição de PMRET para obter mais detalhes sobre as publicações retidas), mas não será usado como o MDAID quando a mensagem for enviada como uma publicação para os assinantes, pois eles fornecem um valor para substituir MDAID em todas as publicações enviadas para eles. Se a mensagem não tiver nenhum contexto, o campo estará inteiramente em branco

Este é um campo de saída para a chamada MQGET O comprimento deste campo é fornecido por LNAIDD O valor inicial desse campo é 32 caracteres em branco.

### **MDAOD (cadeia de caracteres de 4 bytes).**

Dados de aplicativos relacionados à origem.

Isso faz parte do *contexto de origem* da mensagem.. Para obter mais informações sobre o contexto de mensagem, consulte [Contexto da mensagem](#) e [Informações de contexto de controle](#).

MDAOD são informações definidas pelo conjunto de aplicativos que podem ser usadas para fornecer informações adicionais sobre a origem da mensagem. Por exemplo, ele poderia ser configurado por aplicativos em execução com autoridade de usuário adequada para indicar se os dados de identificação são confiáveis..

O gerenciador de filas trata essas informações como dados de caractere, mas não define o formato deles. Quando o gerenciador de filas gera essas informações, ele fica totalmente em branco

Para as chamadas MQPUT e MQPUT1 , este é um campo de entrada /saída se PMSETA for especificado no parâmetro **PMO** . Quaisquer informações após um caractere nulo no campo são descartadas. O caractere nulo e quaisquer caracteres a seguir são convertidos em branco pelo gerenciador de filas. Se o PMSETA não for especificado, esse campo será ignorado na entrada e será um campo somente de saída

Após a conclusão bem-sucedida de uma chamada MQPUT ou MQPUT1 , esse campo contém o MDAOD que foi transmitido com a mensagem se ele foi colocado em uma fila. Esse será o valor de MDAOD que será mantido com a mensagem se ela for retida (consulte a descrição de PMRET para obter mais detalhes sobre as publicações retidas), mas não será usado como o MDAOD quando a mensagem for enviada como uma publicação para os assinantes, pois eles fornecem um valor para substituir MDAOD em todas as publicações enviadas para eles. Se a mensagem não tiver nenhum contexto, o campo estará inteiramente em branco

Este é um campo de saída para a chamada MQGET O comprimento deste campo é fornecido por LNAORD. O valor inicial deste campo é 4 caracteres em branco.

### **MDBOC (número inteiro assinado de 10 dígitos)**

Contador de restauração.

Esta é uma contagem do número de vezes em que a mensagem foi retornada anteriormente pela chamada MQGET como parte de uma unidade de trabalho e, subsequentemente, restaurada Ele é fornecido como um auxílio para o aplicativo na detecção de erros de processamento baseados no conteúdo da mensagem. A contagem exclui as chamadas MQGET que especificaram qualquer uma das opções GMBRW\*

A precisão dessa contagem é afetada pelo atributo da fila **HardenGetBackout** ; consulte [“Atributos para filas”](#) na página 1416.

Este é um campo de saída para a chamada MQGET Ele é ignorado para as chamadas MQPUT e MQPUT1 . O valor inicial desse campo é 0.

## **MDCID (sequência de bits de 24 bytes)**

Identificador de correlação.

Esta é uma cadeia de bytes que o aplicativo pode utilizar para relacionar uma mensagem a outra, ou para relacionar a mensagem a outro trabalho que o aplicativo está executando. O identificador de correlação é uma propriedade permanente da mensagem e persiste em reinicializações do gerenciador de filas. Como o identificador de correlação é uma sequência de bytes e não uma sequência de caracteres, o identificador de correlação não é convertido entre conjuntos de caracteres quando a mensagem flui de um gerenciador de filas para outro.

Para chamadas MQPUT e MQPUT1, o aplicativo pode especificar qualquer valor. O gerenciador de filas transmite esse valor com a mensagem e o entrega ao aplicativo que emite a solicitação get para a mensagem.

Se o aplicativo especificar PMNCID, o gerenciador de filas gerará um identificador de correlação exclusivo que será enviado com a mensagem e também retornado para o aplicativo de envio na saída da chamada MQPUT ou MQPUT1.

Esse identificador de correlação gerado será mantido com a mensagem se ela for retida e for usada como o identificador de correlação quando a mensagem for enviada como uma publicação para assinantes que especificam CINONE no campo SDCID no MQSD transmitido na chamada MQSUB.

Consulte [“MQPMO \(opções Put-message\) em IBM i” na página 1215](#) para obter mais detalhes sobre publicações retidas

Quando o gerenciador de filas ou um agente do canal de mensagens gera uma mensagem de relatório, ele configura o campo MDCID na maneira especificada pelo campo MDREP da mensagem original, ROCMTC ou ROPCI. Os aplicativos que geram mensagens de relatório também devem fazer isso

Para a chamada MQGET, MDCID é um dos cinco campos que podem ser usados para selecionar uma mensagem específica a ser recuperada da fila. Consulte a descrição do campo MDMID para obter detalhes de como especificar valores para esse campo

Especificar CINONE como o identificador de correlação tem o mesmo efeito que não especificar MOCORI, ou seja, qualquer identificador de correlação será correspondente..

Se a opção GMMUC for especificada no parâmetro **GMO** na chamada MQGET, esse campo será ignorado..

No retorno de uma chamada MQGET, o campo MDCID é configurado para o identificador de correlação da mensagem retornada (se houver).

Podem ser utilizados os seguintes valores especiais:

### **CINONE**

Nenhum identificador de correlação é especificado.

O valor é binário zero para o comprimento do campo.

### **CINEWS**

A mensagem é o início de uma nova sessão

Esse valor é reconhecido pelo CICS bridge como indicando o início de uma nova sessão, ou seja o início de uma nova sequência de mensagens.

Para a chamada MQGET, este é um campo de entrada / saída.. Para as chamadas MQPUT e MQPUT1, esse é um campo de entrada se PMNCID não for especificado e um campo de saída se PMNCID for especificado. O comprimento desse campo é fornecido por LNCID. O valor inicial deste campo é CINONE..

## **MDCSI (número inteiro assinado de 10 dígitos).**

Especifica o identificador do conjunto de caracteres de dados de caracteres na mensagem.

**Nota:** Os dados de caractere no MQMD e as outras estruturas de dados do IBM MQ que são parâmetros em chamadas devem estar no conjunto de caracteres do gerenciador de filas Isso

é definido pelo atributo **CodedCharSetId** do gerenciador de filas; consulte “Atributos para o gerenciador de filas no IBM i” na página 1449 para obter detalhes desse atributo..

Os seguintes valores especiais podem ser usados:

### **CSQM**

Identificador do conjunto de caracteres do gerenciador de filas.

Os dados de caracteres na mensagem estão no conjunto de caracteres do gerenciador de fila.

Nas chamadas MQPUT e MQPUT1 , o gerenciador de filas muda esse valor no MQMD enviado com a mensagem para o identificador do conjunto de caractere verdadeiro do gerenciador de filas. Como resultado, o valor CSQM nunca é retornado pela chamada MQGET..

### **CSINHT**

Herde o identificador do conjunto de caracteres desta estrutura..

Os dados de caractere na mensagem estão no mesmo conjunto de caracteres que esta estrutura; este é o conjunto de caracteres do gerenciador de filas (Apenas para MQMD, CSINHT tem o mesmo significado que CSQM)

O gerenciador de filas muda esse valor no MQMD enviado com a mensagem para o identificador do conjunto de caracteres real de MQMD... Se nenhum erro ocorrer, o valor CSINHT não será retornado pela chamada MQGET.

CSINHT não poderá ser usado se o valor do campo MDPAT em MQMD for ATBRKR.

### **CSEMBD**

Identificador do conjunto de caracteres integrado.

Os dados de caractere na mensagem estão em um conjunto de caracteres com o identificador que está contido nos próprios dados da mensagem... Pode haver qualquer número de identificadores de conjunto de caracteres integrados nos dados da mensagem, que se aplicam a diferentes partes dos dados.. Este valor deve ser usado para mensagens PCF que contêm dados em uma mistura de conjuntos de caracteres.. As mensagens PCF têm um nome de formato de FMPCF

Especifique este valor somente nas chamadas MQPUT e MQPUT1 . Se ele for especificado na chamada MQGET, ele evitará a conversão da mensagem

Nas chamadas MQPUT e MQPUT1 , o gerenciador de filas muda os valores CSQM e CSINHT no MQMD enviado com a mensagem conforme descrito anteriormente, mas não muda o MQMD especificado na chamada MQPUT ou MQPUT1 . Nenhuma outra verificação é executada no valor especificado

Os aplicativos que recuperam mensagens devem comparar esse campo com o valor que o aplicativo está esperando; se os valores forem diferentes, o aplicativo poderá precisar converter os dados de caractere na mensagem..

Se a opção GMCONV for especificada na chamada MQGET, esse campo será um campo de entrada e saída O valor especificado pelo aplicativo é o identificador do conjunto de caracteres codificados para o qual os dados da mensagem devem ser convertidos, se necessário. Se a conversão for bem-sucedida ou desnecessária, o valor será inalterado (exceto que o valor CSQM ou CSINHT é convertido para o valor real. Se a conversão for malsucedida, o valor após a chamada MQGET representa o identificador do conjunto de caracteres codificado da mensagem não convertida que é retornada ao aplicativo.

Caso contrário, este é um campo de saída para a chamada MQGET e um campo de entrada para as chamadas MQPUT e MQPUT1 . O valor inicial desse campo é CSQM.

### **MDENC (número inteiro assinado de 10 dígitos)**

Codificação numérica de dados da mensagem.

Especifica a codificação numérica de dados numéricos na mensagem; ela não se aplica a dados numéricos na própria estrutura MQMD. A codificação numérica define a representação usada para números inteiros binários, números inteiros decimais compactados e números de vírgula flutuante..

Na chamada MQPUT ou MQPUT1 , o aplicativo deve configurar esse campo para o valor apropriado aos dados. O gerenciador de filas não verifica se o campo é válido.. O seguinte valor especial é definido:

### **ENNAT**

Codificação de máquina nativa..

A codificação é o padrão para a linguagem de programação e a máquina na qual o aplicativo está em execução

**Nota:** O valor desta constante depende da linguagem de programação e ambiente. Por esse motivo, os aplicativos devem ser compilados usando o cabeçalho, macro, arquivos COPY ou INCLUDE apropriados para o ambiente no qual o aplicativo será executado.

Os aplicativos que colocam mensagens devem normalmente especificar ENNAT. Os aplicativos que recuperam mensagens devem comparar esse campo com o valor ENNAT; se os valores forem diferentes, o aplicativo poderá precisar converter dados numéricos na mensagem... A opção GMCONV pode ser usada para solicitar o gerenciador de fila para converter a mensagem como parte do processamento da chamada MQGET.

Se a opção GMCONV for especificada na chamada MQGET, esse campo será um campo de entrada e saída O valor especificado pelo aplicativo é a codificação para a qual os dados da mensagem devem ser convertidos, se necessário. Se a conversão for bem-sucedida ou desnecessária, o valor será inalterado. Se a conversão for malsucedida, o valor após a chamada MQGET representa a codificação da mensagem não convertida que é retornada ao aplicativo.

Em outros casos, esse é um campo de saída para a chamada MQGET e um campo de entrada para as chamadas MQPUT e MQPUT1 . O valor inicial deste campo é ENNAT

### **MDEXP (número inteiro assinado de 10 dígitos)**

Tempo de vida da mensagem.

Este é um período de tempo expresso em décimos de segundo, definido pelo aplicativo que coloca a mensagem. A mensagem se torna elegível para ser descartada, se não tiver sido removida da fila de destino antes de decorrer este período de tempo.

O valor é decrementado para refletir o tempo que a mensagem gasta na fila de destino e também em qualquer fila de transmissão intermediária se a entrada for para uma fila remota. Ele também pode ser reduzido por agentes do canal de mensagens para refletir os tempos de transmissão, se eles forem significativos. Da mesma forma, um aplicativo que encaminhar essa mensagem para outra fila poderá diminuir o valor, se necessário, se ele tiver retido a mensagem por um tempo significativo.. No entanto, o prazo de expiração é tratado como aproximado e o valor não precisa ser reduzido para refletir pequenos intervalos de tempo.

Quando a mensagem for recuperada por um aplicativo usando a chamada MQGET, o campo MDEXP representará a quantia do tempo de expiração original que ainda permanece..

Após o tempo de expiração de uma mensagem, ela se torna elegível para ser descartada pelo gerenciador de filas. Nas implementações atuais, a mensagem é descartada quando ocorre uma chamada MQGET de procura ou não procura que teria retornado a mensagem se ela ainda não tivesse expirado Por exemplo, uma chamada MQGET não navegada com o campo GMMO em MQGMO configurado como MONONE, a partir de uma fila ordenada FIFO fará com que todas as mensagens expiradas sejam descartadas até a primeira mensagem não expirada Com uma fila ordenada de prioridade, a mesma chamada descartará mensagens expiradas de prioridade mais alta e mensagens de prioridade igual que chegaram na fila antes da primeira mensagem não expirada.

Uma mensagem que expirou nunca é retornada a um aplicativo (por uma procura ou uma chamada MQGET não navegada), portanto, o valor no campo MDEXP do descritor de mensagens após uma chamada MQGET bem-sucedida é maior que zero ou o valor especial EIULIM.

Se uma mensagem for colocada em uma fila remota, ela poderá expirar (e ser descartada) enquanto estiver em uma fila de transmissão intermediária, antes que a mensagem atinja a fila de destino.



Um relatório é gerado quando uma mensagem expirada é descartada, se a mensagem especificou uma das opções de relatório ROEXP\*. Se nenhuma dessas opções for especificada, nenhum relatório será gerado; a mensagem será assumida como não mais relevante após esse período de tempo (talvez porque uma mensagem posterior a substituiu).

Qualquer outro programa que descarte mensagens com base no tempo de expiração também deve enviar uma mensagem de relatório apropriada se uma foi solicitada.

**Nota:**

1. Se uma mensagem for colocada com um tempo MDEXP de zero, a chamada MQPUT ou MQPUT1 falhará com o código de razão RC2013; nenhuma mensagem de relatório será gerada neste caso.
2. Como uma mensagem com um tempo de expiração decorrido pode não ser realmente descartada até mais tarde, pode haver mensagens em uma fila que tenham passado seu tempo de expiração e que, portanto, não sejam elegíveis para recuperação. No entanto, essas mensagens contam para o número de mensagens na fila para todos os propósitos, incluindo acionamento de profundidade.
3. Um relatório de expiração é gerado, se solicitado, quando a mensagem é realmente descartada, não quando se torna elegível para descarte.
4. Descartar uma mensagem expirada e a geração de um relatório de expiração, se solicitado, nunca fazem parte da unidade de trabalho do aplicativo, mesmo se a mensagem foi planejada para descartar como resultado de uma chamada MQGET operando em uma unidade de trabalho.
5. Se uma mensagem quase expirada for recuperada por uma chamada MQGET dentro de uma unidade de trabalho e a unidade de trabalho for restaurada subsequentemente, a mensagem poderá se tornar elegível para ser descartada antes de poder ser recuperada novamente.
6. Se uma mensagem quase expirada for bloqueada por uma chamada MQGET com GMLK, a mensagem poderá se tornar elegível para ser descartada antes que possa ser recuperada por uma chamada MQGET com GMMUC; o código de razão RC2034 será retornado nessa chamada MQGET subsequente se isso acontecer..
7. Quando uma mensagem de solicitação com um tempo de expiração maior que zero é recuperada, o aplicativo pode executar uma das ações a seguir ao enviar a mensagem de resposta:
  - Copie o tempo de expiração restante da mensagem de pedido para a mensagem de resposta
  - Configure o tempo de expiração na mensagem de resposta para um valor explícito maior que zero.
  - Configure o tempo de expiração da mensagem de resposta para EIULIMA ação a ser tomada depende do design do conjunto de aplicativos.. No entanto, a ação padrão para colocar mensagens em uma fila de mensagens não entregues (mensagens não entregues) deve ser preservar o tempo de expiração restante da mensagem e continuar a diminuí-lo.
8. As mensagens do acionador são sempre geradas com EIULIM
9. Uma mensagem (normalmente em uma fila de transmissão) que possui um nome MDFMT de FMXQH possui um segundo descritor de mensagens no MQXQH. Portanto, ele possui dois campos MDEXP associados a ele.. Os seguintes pontos adicionais devem ser observados neste caso:
  - Quando um aplicativo coloca uma mensagem em uma fila remota, o gerenciador de filas coloca a mensagem inicialmente em uma fila de transmissão local e prefixa os dados da mensagem do aplicativo com uma estrutura MQXQH O gerenciador de filas configura os valores dos dois campos MDEXP para serem os mesmos especificados pelo aplicativo.  
  
Se um aplicativo colocar uma mensagem diretamente em uma fila de transmissão local, os dados da mensagem já devem começar com uma estrutura MQXQH e o nome do formato deve ser FMXQH (mas o gerenciador de filas não impinge isso). Nesse caso, o aplicativo não precisa configurar os valores desses dois campos MDEXP para serem os mesmos. (O gerenciador de filas não verifica se o campo MDEXP no MQXQH contém um valor válido ou mesmo se os dados da mensagem são longos o suficiente para incluí-lo.)
  - Quando uma mensagem com um nome MDFMT de FMXQH é recuperada de uma fila (se esta for uma fila normal ou uma fila de transmissão), o gerenciador de filas decreta esses dois



campos MDEXP com o tempo gasto esperando na fila.. Nenhum erro será emitido se os dados da mensagem não forem longos o suficiente para incluir o campo MDEXP no MQXQH

- O gerenciador de filas utiliza o campo MDEXP no descritor de mensagem separado (ou seja, não aquele no descritor de mensagem integrado na estrutura MQXQH) para testar se a mensagem é elegível para descarte...
- Se os valores iniciais dos dois campos MDEXP eram diferentes, portanto, é possível que o tempo MDEXP no descritor de mensagens separado quando a mensagem é recuperada seja maior que zero (portanto, a mensagem não é elegível para descarte), enquanto o tempo de acordo com o campo MDEXP no MQXQH foi decorrido. Nesse caso, o campo MDEXP no MQXQH é configurado como zero.

O valor especial a seguir é reconhecido:

#### **EIULIM**

Vida ilimitada.

A mensagem tem um prazo de expiração ilimitado

Esse é um campo de saída para a chamada MQGET e um campo de entrada para as chamadas MQPUT e MQPUT1 . O valor inicial deste campo é EIULIM.

#### **MDFB (número inteiro assinado de 10 dígitos)**

Fornecimento de informações ou código de retorno.

Isso é usado com uma mensagem do tipo MTRPRT para indicar a natureza do relatório e é significativo somente com esse tipo de mensagem O campo pode conter um dos valores FB\* ou um dos valores RC\*. Os códigos de feedback são agrupados da seguinte forma:

#### **FBNONE**

Nenhum feedback fornecido.

#### **FBSFST**

Valor mais baixo para feedback gerado pelo sistema.

#### **FBSLST**

Valor mais alto para feedback gerado pelo sistema.

O intervalo de códigos de feedback gerados pelo sistema FBSFST até FBSLST inclui os códigos de feedback gerais listados posteriormente nesta seção (FB\*) e também os códigos de razão (RC\*) que podem ocorrer quando a mensagem não pode ser colocada na fila de destino.

#### **FBAFST**

Valor mais baixo para feedback gerado pelo aplicativo.

#### **FBALST**

Valor mais alto para feedback gerado pelo aplicativo.

Os aplicativos que geram mensagens de relatório não devem usar códigos de feedback no intervalo do sistema (além de FBQUIT), a menos que desejem simular mensagens de relatório geradas pelo gerenciador de filas ou agente do canal de mensagem.

Nas chamadas MQPUT ou MQPUT1 , o valor especificado deve ser FBNONE ou estar dentro do intervalo do sistema ou intervalo do aplicativo. Isso é verificado independentemente do valor de MDMT.

#### **Códigos de feedback gerais:**

#### **FBCOA**

Confirmação de chegada na fila de destino (ver ROCOA).

#### **FBCOD**

Confirmação da entrega ao pedido de recepção (ver ROCOD).

#### **FBEXP**

Mensagem expirada.

A mensagem foi descartada porque ela não tinha sido removida da fila de destino antes de seu tempo de expiração ter decorrido

**FBPAN**

Notificação de ação positiva (ver ROPAN).

**FBNAN**

Notificação de ação negativa (ver RONAN).

**FBQUIT**

O aplicativo deve terminar.

Isso pode ser usado por um programa de planejamento de carga de trabalho para controlar o número de instâncias de um programa de aplicativo que estão em execução. Enviar uma mensagem MTRPRT com este código de feedback para uma instância do programa de aplicativo indica a essa instância que ela deve parar o processamento. No entanto, a adesão a essa Convenção é uma questão para o aplicativo; ela não é imposta pelo gerenciador de filas

**IMS-bridge feedback codes:** quando a ponte IMS recebe um código de controle diferente de zero IMS-OTMA, a ponte IMS converte o código de controle de hexadecimal para decimal, inclui o valor FBIERR (300) e coloca o resultado no campo MDFB da mensagem de resposta... Isso faz com que o código de feedback tenha um valor no intervalo de FBIFST (301) a FBILST (399) quando ocorreu um erro IMS-OTMA.

Os códigos de feedback a seguir podem ser gerados pela ponte IMS :

**FBDLZ**

Comprimento de dados zero.

Um comprimento de segmento era zero nos dados do aplicativo da mensagem..

**FBDLN**

Comprimento de dados negativo.

Um comprimento do segmento foi negativo nos dados do aplicativo da mensagem

**FBDLTB**

Comprimento de dados muito grande.

Um comprimento de segmento era muito grande nos dados do aplicativo da mensagem..

**FBBUFO**

Estouro de buffer.

O valor de um dos campos de comprimento faria com que os dados sobrecarregassem o buffer de mensagens..

**FBLOB1**

Comprimento em erro por um.

O valor de um dos campos de comprimento era um byte muito curto.

**FBIIH**

A estrutura MQIIH não é válida ou está ausente

O campo MDFMT no MQMD especifica FMIMS, mas a mensagem não começa com uma estrutura MQIIH válida

**FBNAFI**

ID do usuário não autorizado para uso em IMS.

O ID do usuário contido no descritor de mensagens MQMD ou a senha contida no campo IIAUT na estrutura MQIIH falhou na validação executada pela ponte IMS . Como resultado, a mensagem não foi transmitida para o IMS

**FBIERR**

Erro inesperado retornado por IMS.

Um erro inesperado foi retornado por IMS Consulte o log de erro do IBM MQ no sistema no qual a ponte IMS reside para obter mais informações sobre o erro

**FBIFST**

Valor mais baixo para feedback gerado pelo IMS.

IMS-os códigos de feedback gerados ocupam o intervalo FBIFST (300) a FBILST (399). O próprio código de detecção do IMS-OTMA é MDFB menos FBIERR

**FBILST**

Valor mais alto para o feedback gerado pelo IMS.

**CICS-bridge feedback codes:** os códigos de feedback a seguir podem ser gerados pelo CICS bridge:

**FBCAAB**

O aplicativo foi encerrado de forma anormal

O programa de aplicação especificado na mensagem foi encerrado de forma anormal Esse código de feedback ocorre apenas no campo DLREA da estrutura MQDLH..

**FBCANS**

O aplicativo não pode ser iniciado..

O EXEC CICS LINK para o programa de aplicativo especificado na mensagem falhou Esse código de feedback ocorre apenas no campo DLREA da estrutura MQDLH..

**FBCBRF**

CICS bridge finalizou de forma anormal sem concluir o processamento de erro normal.

**FBCCSSE**

O identificador do conjunto de caracteres não é válido.

**FBCIHE**

CICS estrutura do cabeçalho de informações ausente ou inválida.

**FBCCAE**

O comprimento de CICS commarea não é válido

**FBCIE**

Identificador de correlação inválido.

**FBCDLQ**

Fila de devoluções não disponível.

A tarefa CICS bridge não pôde copiar uma resposta para essa solicitação para a fila de mensagens não entregues A solicitação foi restaurada..

**FBCENE**

A codificação não é válida

**FBCINE**

CICS bridge encontrou um erro inesperado.

Esse código de feedback ocorre apenas no campo DLREA da estrutura MQDLH..

**FBCNTA**

Identificador de usuário não autorizado ou senha inválida.

Esse código de feedback ocorre apenas no campo DLREA da estrutura MQDLH..

**FBCUBO**

Unidade de trabalho restaurada.

A unidade de trabalho foi restaurada, por uma das seguintes razões:

- Uma falha foi detectada ao processar outra solicitação dentro da mesma unidade de trabalho.
- Ocorreu um encerramento anormal CICS enquanto a unidade de trabalho estava em andamento.

**FBCUWE**

O campo de controle da unidade de trabalho CIUOW não é válido

**MQ códigos de razão:** para mensagens de relatório de exceção, MDFB contém um código de razão MQ . Entre os possíveis códigos de razão estão:

**RC2051**

(2051, X'803 ') Chamadas de put inibidas para a fila

**RC2053**

(2053, X'805 ') A fila já contém número máximo de mensagens.

**RC2035**

(2035, X'7F3') Não autorizado para acesso.

**RC2056**

(2056, X'808 ') Nenhum espaço disponível no disco para fila.

**RC2048**

(2048, X'800 ') A Fila não suporta mensagens persistentes.

**RC2031**

(2031, X'7EF') Comprimento da mensagem maior que o máximo para o gerenciador de filas.

**RC2030**

(2030, X'7EE') Comprimento da mensagem maior que o máximo para a fila.

Esse é um campo de saída para a chamada MQGET e um campo de entrada para as chamadas MQPUT e MQPUT1 . O valor inicial deste campo é FBNONE.

**MDFMT (sequência de caracteres de 8 bytes).**

Nome do formato dos dados da mensagem.

Este é um nome que o emissor da mensagem pode utilizar para indicar ao receptor a natureza dos dados na mensagem. Quaisquer caracteres que estejam no conjunto de caracteres do gerenciador de filas podem ser especificados para o nome, mas é recomendado que o nome seja restrito ao seguinte:

- Maiúscula A a Z
- Dígitos numéricos de 0 a 9

Se outros caracteres forem usados, talvez não seja possível converter o nome entre os conjuntos de caracteres dos gerenciadores de fila de envio e de recebimento...

O nome deve ser preenchido com espaços em branco para o comprimento do campo ou um caractere nulo utilizado para finalizar o nome antes do final do campo; o nulo e quaisquer caracteres subsequentes são tratados como espaços em branco. Não especifique um nome com espaços em branco iniciais ou integrados. Para a chamada MQGET, o gerenciador de filas retorna o nome preenchido com espaços em branco para o comprimento do campo

O gerenciador de filas não verifica se o nome está em conformidade com as recomendações descritas anteriormente

Os nomes que começam com "MQ" em maiúsculas, minúsculas e combinadas têm significados que são definidos pelo gerenciador de filas; você não deve usar nomes que começam com essas letras para seus próprios formatos. Os formatos integrados do gerenciador de filas são:

**FMNONE**

Nome sem formato.

A natureza dos dados é indefinida Isto significa que os dados não podem ser convertidos quando a mensagem é recuperada de uma fila utilizando a opção GMCONV.

Se GMCONV for especificado na chamada MQGET e o conjunto de caracteres ou a codificação de dados na mensagem for diferente daquele especificado no parâmetro **MSGDSC** , a mensagem será retornada com os seguintes códigos de conclusão e de razão (assumindo que nenhum outro erro):

- Código de conclusão CCWARN e código de razão RC2110 se os dados FMNONE estiverem no início da mensagem.

- Código de conclusão CCOK e código de razão RCNONE se os dados FMNONE estiverem no final da mensagem (ou seja, precedidos por uma ou mais estruturas de cabeçalho MQ ). As estruturas do cabeçalho MQ são convertidas para o conjunto de caracteres solicitado e a codificação nesse caso.

#### **FMADMN**

Mensagem de pedido/resposta do servidor de comandos.

A mensagem é uma mensagem de solicitação ou resposta do servidor de comando no formato de comando programável (PCF). As mensagens desse formato poderão ser convertidas se a opção GMCONV for especificada na chamada MQGET Para obter mais informações sobre como usar mensagens de formato de comando programável, consulte [Usando Formatos de Comando Programáveis](#).

#### **DECICS**

Cabeçalho de informações do CICS

Os dados da mensagem começam com o cabeçalho de informações do CICS MQCIH, que é seguido pelos dados do aplicativo O nome do formato dos dados do aplicativo é fornecido pelo campo CIFMT na estrutura MQCIH.

#### **FMCMD1**

Mensagem de resposta do comando tipo 1.

A mensagem é uma mensagem de resposta do servidor de comandos MQSC que contém a contagem de objetos, o código de conclusão e o código de razão As mensagens desse formato poderão ser convertidas se a opção GMCONV for especificada na chamada MQGET

#### **FMCMD2**

Mensagem de resposta do comando tipo 2.

A mensagem é uma mensagem de resposta do servidor de comandos MQSC contendo informações sobre o (s) objeto (s) solicitado (s). As mensagens desse formato poderão ser convertidas se a opção GMCONV for especificada na chamada MQGET

#### **FMDLH**

Cabeçalho da fila de devoluções.

Os dados da mensagem começam com o cabeçalho MQDLH. Os dados da mensagem original seguem imediatamente a estrutura MQDLH.. O nome do formato dos dados da mensagem original é fornecido por o campo DLFMT na estrutura MQDLH; consulte [“MQDLH \(cabeçalho de devoluções\) em IBM i” na página 1101](#) para obter detalhes dessa estrutura As mensagens desse formato poderão ser convertidas se a opção GMCONV for especificada na chamada MQGET

Os relatórios de COA e COD não são gerados para mensagens que possuem um MDFMT de FMDLH

#### **FMDH**

Cabeçalho da lista de distribuições.

Os dados da mensagem começam com o cabeçalho MQDH da lista de distribuição; isso inclui as matrizes de registros MQOR e MQPMR. O cabeçalho da lista de distribuição pode ser seguido por dados adicionais.. O formato dos dados adicionais (se houver) é fornecido pelo campo DHFMT na estrutura MQDH; consulte [“MQDH \(Cabeçalho de distribuição\) no IBM i” na página 1096](#) para obter detalhes dessa estrutura. Mensagens com formato FMDH podem ser convertidas se a opção GMCONV for especificada na chamada MQGET.

#### **FMEVNT**

Mensagem do evento.

A mensagem é uma mensagem do evento MQ que relata um evento ocorrido. As mensagens de eventos possuem a mesma estrutura que os comandos programáveis; para obter mais informações sobre essa estrutura, consulte [Estruturas para comandos e respostas](#). Para obter informações sobre eventos, consulte [Monitoramento de Eventos](#)..

As mensagens do evento Version-1 poderão ser convertidas se a opção GMCONV for especificada na chamada MQGET.

## **DEIMS**

Cabeçalho de informações do IMS

Os dados da mensagem começam com o cabeçalho de informações MQIIH do IMS , que é seguido pelos dados do aplicativo O nome do formato dos dados do aplicativo é fornecido pelo campo *IIFMT* na estrutura MQIIH. As mensagens desse formato poderão ser convertidas se a opção GMCONV for especificada na chamada MQGET

## **FMIMVS**

Sequência de variáveis IMS

A mensagem é uma sequência de variáveis IMS , que é uma sequência do formato 11zzccc, em que:

### **11**

é um campo de comprimento de 2 bytes especificando o comprimento total do item de sequência variável IMS . Esse comprimento é igual ao comprimento de 11 (2 bytes), mais o comprimento de zz (2 bytes), mais o comprimento da própria cadeia de caractere. 11 é um número inteiro binário de 2 bytes na codificação especificada pelo campo MDENC .

### **zz**

é um campo de 2 bytes contendo sinalizadores que são significativos para IMS. zz é uma sequência de bytes que consiste em dois campos de sequência de bits de 1 byte e é transmitida sem mudança de emissor para receptor (ou seja, zz não está sujeito a nenhuma conversão).

### **ccc**

é uma sequência de caracteres de comprimento variável que contém 11-4 caracteres. ccc está no conjunto de caracteres especificado pelo campo MDCSI ..

As mensagens desse formato poderão ser convertidas se a opção GMCONV for especificada na chamada MQGET

## **FMMDE**

Extensão do descritor de mensagem

Os dados da mensagem começam com a extensão do descritor de mensagens MQMDE e são opcionalmente seguidos por outros dados (geralmente os dados da mensagem do aplicativo). O nome do formato, o conjunto de caracteres e a codificação dos dados que seguem o MQMDE são fornecidos pelos campos MEFMT, MECSIE MEENC no MQMDE Consulte [“MQMDE \(extensão do descritor de mensagens\) em IBM i”](#) na página 1194 para obter detalhes sobre essa estrutura As mensagens desse formato poderão ser convertidas se a opção GMCONV for especificada na chamada MQGET

## **FMPCF**

Mensagem definida pelo usuário no formato de comando programável (PCF)..

A mensagem é uma mensagem definida pelo usuário em conformidade com a estrutura de uma mensagem de formato de comando programável (PCF). As mensagens desse formato poderão ser convertidas se a opção GMCONV for especificada na chamada MQGET Consulte [Usando Formatos de Comando Programáveis](#) para obter mais informações sobre como usar mensagens de formato de comando programáveis

## **FMRMH**

Cabeçalho da mensagem de referência..

Os dados da mensagem começam com o cabeçalho da mensagem de referência MQRMH e são opcionalmente seguidos por outros dados. O nome do formato, o conjunto de caracteres e a codificação dos dados são fornecidos pelos campos RMFMT, RMCSIE RMENC no MQRMH Consulte [“MQRMH \(Cabeçalho da mensagem de referência\) no IBM i”](#) na página 1243 para obter detalhes sobre essa estrutura As mensagens desse formato poderão ser convertidas se a opção GMCONV for especificada na chamada MQGET

## **FMRFH**

Regras e cabeçalho de formatação..

Os dados da mensagem começam com as regras e o cabeçalho de formatação MQRFH e são opcionalmente seguidos por outros dados. O nome do formato, o conjunto de caracteres e a codificação dos dados (se houver) são fornecidos pelos campos RFFMT, RFCSIe RFENC no MQRFH. As mensagens desse formato poderão ser convertidas se a opção GMCONV for especificada na chamada MQGET

#### **FMRFH2**

Regras e cabeçalho de formatação versão 2.

Os dados da mensagem começam com as regras version-2 e o cabeçalho de formatação MQRFH2e, opcionalmente, são seguidos por outros dados. O nome do formato, o conjunto de caracteres e a codificação dos dados opcionais (se houver) são fornecidos pelos campos RF2FMT, RF2CSIe RF2ENC no MQRFH2. As mensagens desse formato poderão ser convertidas se a opção GMCONV for especificada na chamada MQGET

#### **FMSTR**

Mensagem consistindo totalmente em caracteres.

Os dados da mensagem do aplicativo podem ser uma sequência SBCS (conjunto de caracteres de byte único) ou uma sequência DBCS (conjunto de caracteres de byte duplo). As mensagens desse formato poderão ser convertidas se a opção GMCONV for especificada na chamada MQGET

#### **FMTM**

Mensagem do acionador..

A mensagem é uma mensagem acionador, descrita pela estrutura MQTM; consulte [“MQTM- Mensagem do acionador”](#) na página 1280 para obter detalhes dessa estrutura. As mensagens desse formato poderão ser convertidas se a opção GMCONV for especificada na chamada MQGET

#### **FMWIH**

Cabeçalho de informações de trabalho

Os dados da mensagem começam com o cabeçalho de informações de trabalho MQWIH, seguido pelos dados do aplicativo. O nome do formato dos dados do aplicativo é fornecido pelo campo WIFMT na estrutura MQWIH.

#### **FMXQH**

Cabeçalho da fila de transmissão.

Os dados da mensagem começam com o cabeçalho da fila de transmissão MQXQH.. Os dados da mensagem original seguem imediatamente a estrutura MQXQH.. O nome do formato dos dados da mensagem original é fornecido pelo campo MDFMT na estrutura MQMD que faz parte do MQXQH do cabeçalho da fila de transmissão Consulte [“MQXQH \(Transmission-queue header\) em IBM i”](#) na página 1290 para obter detalhes sobre essa estrutura

Os relatórios COA e COD não são gerados para mensagens que possuem um MDFMT de FMXQH

Esse é um campo de saída para a chamada MQGET e um campo de entrada para as chamadas MQPUT e MQPUT1 . O comprimento deste campo é fornecido pelo LNFMT O valor inicial deste campo é FMNONE.

#### **MDGID (sequência de bits de 24 bytes)**

Identificador de grupo.

Esta é uma cadeia de bytes que é utilizada para identificar o grupo de mensagens específico ou a mensagem lógica à qual a mensagem física pertence MDGID também será usado se a segmentação for permitida para a mensagem Em todos esses casos, MDGID possui um valor não nulo e um ou mais dos sinalizadores a seguir está configurado no campo MDMFL :

- MFMIG
- MFLMIG
- MFSEG
- MFLSEG

- MFSEGA

Se nenhum desses sinalizadores for configurado, o MDGID terá o valor nulo especial GINONE.

Este campo não precisa ser configurado pelo aplicativo na chamada MQPUT ou MQGET se:

- Na chamada MQPUT, PMLOGO é especificado..
- Na chamada MQGET, MOGRPI não é especificado..

Considere o uso dessas chamadas para mensagens que não são mensagens de relatório. No entanto, se o aplicativo requerer mais controle ou a chamada for MQPUT1, o aplicativo deverá assegurar que MDGID seja configurado para um valor apropriado.

Os grupos de mensagens e segmentos poderão ser processados corretamente apenas se o identificador de grupo for exclusivo... Por esta razão, os aplicativos não devem gerar seus próprios identificadores de grupo; em vez disso, os aplicativos devem executar um dos seguintes:

- Se PMLOGO for especificado, o gerenciador de fila gerará automaticamente um identificador de grupo exclusivo para a primeira mensagem no grupo ou segmento da mensagem lógica e usará esse identificador de grupo para as mensagens restantes no grupo ou nos segmentos da mensagem lógica, portanto, o aplicativo não precisa executar nenhuma ação especial. Considere o uso deste procedimento
- Se PMLOGO não for especificado, o aplicativo deverá solicitar que o gerenciador de filas gere o identificador de grupo, configurando MDGID como GINONE na primeira chamada MQPUT ou MQPUT1 para uma mensagem no grupo ou segmento da mensagem lógica. O identificador de grupo retornado pelo gerenciador de filas na saída dessa chamada deve então ser usado para as mensagens restantes no grupo ou segmentos da mensagem lógica. Se um grupo de mensagens contiver mensagens segmentadas, o mesmo identificador de grupo deverá ser usado para todos os segmentos e mensagens no grupo..

Quando PMLOGO não é especificado, as mensagens em grupos e segmentos de mensagens lógicas podem ser colocadas em qualquer ordem (por exemplo, em ordem reversa), mas o identificador de grupo deve ser alocado pela primeira chamada MQPUT ou MQPUT1 que é emitida para qualquer uma dessas mensagens.

Na entrada para as chamadas MQPUT e MQPUT1, o gerenciador de filas usa o valor detalhado em PMOPT. Na saída das chamadas MQPUT e MQPUT1, o gerenciador de filas configura esse campo para o valor que foi enviado com a mensagem se o objeto aberto for uma única fila e não uma lista de distribuição, mas o deixa inalterado se o objeto aberto for uma lista de distribuição. No último caso, se o aplicativo precisar saber os identificadores de grupo gerados, o aplicativo deverá fornecer registros MQPMR contendo o campo PRGID ..

Na entrada para a chamada MQGET, o gerenciador de filas usa o valor detalhado na Tabela 1. Na saída da chamada MQGET, o gerenciador de filas configura esse campo para o valor da mensagem recuperada.

O seguinte valor especial é definido:

#### **GINONE**

Nenhum identificador de grupo especificado.

O valor é binário zero para o comprimento do campo. Esse é o valor usado para mensagens que não estão em grupos, não segmentos de mensagens lógicas e para as quais a segmentação não é permitida.

O comprimento deste campo é fornecido pelo LNGID. O valor inicial deste campo é GINONE. Esse campo será ignorado se MDVER for menor que MDVER2.

#### **MDMFL (número inteiro assinado de 10 dígitos)**

Sinalizadores de mensagem

Esses são sinalizadores que especificam atributos da mensagem ou controlam seu processamento. Os sinalizadores são divididos nas seguintes categorias:



- Indicador segmentação
- Sinalizações de status

Estes são descritos por sua vez

**Sinalização de Segmentação:** Quando uma mensagem é muito grande para uma fila, uma tentativa de colocar a mensagem na fila geralmente falha. A segmentação é uma técnica pela qual o gerenciador de filas ou aplicativo divide a mensagem em partes menores chamadas segmentos e coloca cada segmento na fila como uma mensagem física separada. O aplicativo que recupera a mensagem pode recuperar os segmentos um por um ou solicitar ao gerenciador de filas para remontar os segmentos em uma única mensagem que é retornada por uma chamada MQGET. O último é obtido especificando a opção GMCMPM na chamada MQGET, e fornecendo um buffer que é grande o suficiente para acomodar a mensagem completa (Consulte [“MQGMO \(opções GetMessage\) em IBM i”](#) na página 1113 para obter detalhes da opção GMCMPM.) A segmentação de uma mensagem pode ocorrer no gerenciador de filas de envio, em um gerenciador de filas intermediário, ou no gerenciador de filas de destino

É possível especificar um dos seguintes para controlar a segmentação de uma mensagem:

#### **MFSEGI**

Segmentação inibida.

Essa opção impede que a mensagem seja dividida em segmentos pelo gerenciador de filas. Se especificado para uma mensagem que já é um segmento, essa opção impede que o segmento seja dividido em segmentos menores.

O valor desse sinalizador é binário zero. Esse é o padrão.

#### **MFSEGA**

Segmentação permitida.

Esta opção permite que a mensagem seja dividida em segmentos pelo Gerenciador de Filas. Se especificado para uma mensagem que já é um segmento, esta opção permite que o segmento seja dividido em segmentos menores.. MFSEGA pode ser configurado sem MFSEG ou MFLSEG sendo configurado.

Quando o gerenciador de filas segmenta uma mensagem, o gerenciador de filas ativa a sinalização MFSEG na cópia do MQMD que é enviada com cada segmento, mas não altera as configurações dessas sinalizações no MQMD fornecido pelo aplicativo na chamada MQPUT ou MQPUT1 . Para o último segmento na mensagem lógica, o gerenciador de filas também ativa o sinalizador MFLSEG no MQMD enviado com o segmento.

**Nota:** É necessário cuidado quando as mensagens são colocadas com MFSEGA, mas sem PMLOGO. Se a mensagem for:

- Não é um segmento, e
- Não em um grupo, e
- Não sendo encaminhado,

o aplicativo deve se lembrar de reconfigurar o campo MDGID para GINONE antes de cada chamada MQPUT ou MQPUT1 , para fazer com que um identificador de grupo exclusivo seja gerado pelo gerenciador de filas para cada mensagem. Se isso não for feito, as mensagens não relacionadas poderão terminar inadvertidamente com o mesmo identificador de grupo, o que pode levar a um processamento incorreto subsequentemente Consulte as descrições do campo MDGID e a opção PMLOGO para obter mais informações sobre quando o campo MDGID deve ser reconfigurado

O gerenciador de filas divide as mensagens em segmentos conforme necessário para assegurar que os segmentos (mais quaisquer dados de cabeçalho que possam ser necessários) se ajustem na fila. No entanto, há um limite inferior para o tamanho de um segmento gerado pelo gerenciador de filas e somente o último segmento criado a partir de uma mensagem pode ser menor que esse limite. (O limite inferior para o tamanho de um segmento gerado pelo aplicativo é um byte.) Os

segmentos gerados pelo gerenciador de filas podem ter comprimento diferente. O gerenciador de filas processa a mensagem como segue:

- Formatos definidos pelo usuário são divididos em limites que são múltiplos de 16 bytes. Isso significa que o gerenciador de fila não gerará segmentos menores que 16 bytes (diferentes do último segmento).
- Formatos integrados diferentes de FMSTR são divididos em pontos apropriados à natureza dos dados presentes. No entanto, o gerenciador de filas nunca divide uma mensagem no meio de uma estrutura do cabeçalho MQ . Isso significa que um segmento contendo uma estrutura de cabeçalho MQ única não pode ser dividido ainda mais pelo gerenciador de filas e, como resultado, o tamanho mínimo possível do segmento para essa mensagem é maior que 16 bytes.

O segundo segmento ou posterior gerado pelo gerenciador de filas começará com um dos seguintes:

- Uma estrutura do cabeçalho MQ
  - O início dos dados da mensagem do aplicativo
  - Meio caminho através dos dados da mensagem do aplicativo
- FMSTR é dividido sem considerar a natureza dos dados presentes (SBCS, DBCS ou SBCS/DBCS misto). Quando a cadeia é DBCS ou SBCS/DBCS misto, isto pode resultar em segmentos que não podem ser convertidos de um conjunto de caracteres para outro.. O gerenciador de filas nunca divide mensagens FMSTR em segmentos menores que 16 bytes (diferentes do último segmento).
  - Os campos MDFMT, MDCSI e MDENC no MQMD de cada segmento são configurados pelo gerenciador de filas para descrever corretamente os dados presentes no início do segmento; o nome do formato será o nome de um formato integrado, ou o nome de um formato definido pelo usuário
  - O campo MDREP no MQMD de segmentos com MDOFF maior que zero é modificado conforme a seguir:
    - Para cada tipo de relatório, se a opção de relatório for RO\* D, mas o segmento não puder conter nenhum dos primeiros 100 bytes de dados do usuário (ou seja, os dados após quaisquer estruturas de cabeçalho do MQ que possam estar presentes), a opção de relatório será alterada para RO\*.

O gerenciador de filas segue as regras anteriores, mas, caso contrário, divide mensagens imprevisivelmente; não faça suposições sobre onde uma mensagem é dividida

Para mensagens persistentes, o gerenciador de filas pode executar a segmentação apenas dentro de uma unidade de trabalho:

- Se a chamada MQPUT ou MQPUT1 estiver operando em uma unidade de trabalho definida pelo usuário, essa unidade de trabalho será usada. Se a chamada falhar parcialmente através do processo de segmentação, o gerenciador de filas remove quaisquer segmentos que foram colocados na fila como resultado da chamada com falha. Entretanto, a falha não impede que a unidade de trabalho seja confirmada com êxito.
- Se a chamada estiver operando fora de uma unidade de trabalho definida pelo usuário e não houver nenhuma unidade de trabalho definida pelo usuário existente, o gerenciador de filas criará uma unidade de trabalho apenas pela duração da chamada. Se a chamada for bem-sucedida, o gerenciador de fila confirma a unidade de trabalho automaticamente (o aplicativo não precisa fazer isso). Se a chamada falhar, o gerenciador de filas volta a unidade de trabalho.
- Se a chamada estiver operando fora de uma unidade de trabalho definida pelo usuário, mas uma unidade de trabalho definida pelo usuário existir, o gerenciador de filas não poderá executar a segmentação. Se a mensagem não precisar de segmentação, a chamada ainda poderá ser bem-sucedida Mas se a mensagem não requer segmentação, a chamada falha com o código de razão RC2255.

Para mensagens não persistentes, o gerenciador de filas não requer que uma unidade de trabalho esteja disponível para executar a segmentação.

Deve ser dada especial atenção à conversão de dados de mensagens que possam ser segmentadas:

- Se a conversão de dados for executada somente por o aplicativo de recebimento na chamada MQGET e o aplicativo especificar a opção GMCMPM, a saída de conversão de dados será transmitida a mensagem completa para a saída para conversão e o fato de que a mensagem foi segmentada não será aparente para a saída.
- Se o aplicativo receptor recuperar um segmento por vez, a saída de conversão de dados será chamada para converter um segmento por vez. Portanto, a saída deve ser capaz de converter os dados em um segmento independentemente dos dados em qualquer um dos outros segmentos.

Se a natureza dos dados na mensagem for tal que a segmentação arbitrária dos dados em limites de 16 bytes pode resultar em segmentos que não podem ser convertidos pela saída, ou o formato é FMSTR e o conjunto de caracteres é DBCS ou misto SBCS/DBCS, o próprio aplicativo de envio deve criar e colocar os segmentos, especificando MFSEGI para suprimir segmentação adicional. Dessa forma, o aplicativo de envio pode assegurar que cada segmento contenha informações suficientes para permitir que a saída de conversão de dados converta o segmento com êxito

- Se a conversão do emissor for especificada para um agente do canal de mensagens de envio (MCA), o MCA converterá apenas mensagens que não sejam segmentos de mensagens lógicas; o MCA nunca tenta converter mensagens que sejam segmentos.

Esse sinalizador é uma sinalização de entrada nas chamadas MQPUT e MQPUT1 e uma sinalização de saída na chamada MQGET. Na última chamada, o gerenciador de fila também ecoa o valor da sinalização para o campo GMSEG em MQGMO.

O valor inicial dessa sinalização é MFSEGI.

**Sinalizadores de status:** Estes são sinalizadores que indicam se a mensagem física pertence a um grupo de mensagens, é um segmento de uma mensagem lógica, ambos ou nenhum. Um ou mais dos seguintes podem ser especificados na chamada MQPUT ou MQPUT1 ou retornados pela chamada MQGET:

#### **MFMIIG**

A mensagem é um membro de um grupo

#### **MFLMIIG**

A mensagem é a última mensagem lógica em um grupo

Se esse sinalizador for configurado, o gerenciador de filas ativará MFMIIG na cópia do MQMD que é enviada com a mensagem, mas não altera as configurações desses sinalizadores no MQMD fornecido pelo aplicativo na chamada MQPUT ou MQPUT1 .

É válido que um grupo consista em apenas uma mensagem lógica Se esse for o caso, MFLMIIG será configurado, mas o campo MDSEQ terá o valor um.

#### **MFSEG**

A mensagem é um segmento de uma mensagem lógica

Quando MFSEG é especificado sem MFLSEG, o comprimento dos dados da mensagem do aplicativo no segmento (excluindo os comprimento de quaisquer estruturas de cabeçalho do MQ que possam estar presentes) deve ser pelo menos um. Se o comprimento for zero, a chamada MQPUT ou MQPUT1 falhará com o código de razão RC2253.

#### **MFLSEG**

A mensagem é o último segmento de uma mensagem lógica

Se essa sinalização for configurada, o gerenciador de filas ativará o MFSEG na cópia do MQMD que é enviada com a mensagem, mas não alterará as configurações dessas sinalizações no MQMD fornecido pelo aplicativo na chamada MQPUT ou MQPUT1 .

É válido para uma mensagem lógica consistir em apenas um segmento. Se este for o caso, MFLSEG é configurado, mas o campo MDOFF tem o valor zero.

Quando MFLSEG for especificado, é permitido que o comprimento dos dados da mensagem do aplicativo no segmento (excluindo os comprimentos de quaisquer estruturas de cabeçalho que possam estar presentes) seja zero..

O aplicativo deve assegurar que essas sinalizações sejam configuradas corretamente ao colocar mensagens. Se PMLOGO for especificado ou foi especificado na chamada MQPUT anterior para o identificador de fila, as configurações dos sinalizadores devem ser consistentes com as informações do grupo e do segmento retidas pelo gerenciador de filas para o identificador de fila. As condições a seguir se aplicam a chamadas MQPUT sucessivas para o identificador de fila quando PMLOGO é especificado:

- Se não houver nenhum grupo atual ou mensagem lógica, todas essas sinalizações (e combinações delas) serão válidas.
- Depois que MFMIG tiver sido especificado, ele deverá permanecer ligado até que MFLMIG seja especificado. A chamada falha com o código de razão RC2241 se essa condição não for satisfeita.
- Depois que MFSEG tiver sido especificado, ele deverá permanecer ligado até que MFLSEG seja especificado. A chamada falhará com o código de razão RC2242 se essa condição não for satisfeita.
- Uma vez que MFSEG tenha sido especificado sem MFMIG, MFMIG deve permanecer desativado até que MFLSEG tenha sido especificado. A chamada falhará com o código de razão RC2242 se essa condição não for satisfeita.

Tabela 1 mostra as combinações válidas dos sinalizadores e os valores usados para vários campos.

Esses sinalizadores são sinalizadores de entrada nas chamadas MQPUT e MQPUT1 e sinalizadores de saída na chamada MQGET. Na última chamada, o gerenciador de filas também ecoa os valores dos sinalizadores para os campos GMGST e GMSST em MQGMO

**Sinalizadores padrão:** O seguinte pode ser especificado para indicar que a mensagem possui atributos padrão:

#### **MFNONE**

Nenhum sinalizador de mensagem (atributos de mensagem padrão).

Isso inibe a segmentação, e indica que a mensagem não está em um grupo e não é um segmento de uma mensagem lógica. O MFNONE é definido para auxiliar a documentação do programa. Não se pretende que esse sinalizador seja usado com qualquer outro, mas como seu valor é zero, tal uso não pode ser detectado.

O campo MDMFL é particionado em subcampos; para obter mais detalhes, consulte [“Opções de relatório e sinalizações de mensagem no IBM i”](#) na página 1484

O valor inicial deste campo é MFNONE. Esse campo será ignorado se MDVER for menor que MDVER2.

#### **MDMID (cadeia de bits de 24 bytes)**

Identificador de mensagem.

Esta é uma cadeia de bytes utilizada para distinguir uma mensagem de outra. Geralmente, duas mensagens não devem ter o mesmo identificador de mensagem, embora isso não seja desaprovado pelo gerenciador de filas. O identificador de mensagem é uma propriedade permanente da mensagem e persiste em reinicializações do gerenciador de filas. Como o identificador de mensagem é uma cadeia de bytes e não uma cadeia de caracteres, o identificador de mensagens não é convertido entre conjuntos de caractere quando a mensagem flui de um gerenciador de filas para outro...

Para as chamadas MQPUT e MQPUT1, se MINONE ou PMNMID for especificado pelo aplicativo, o gerenciador de filas gerará um identificador de mensagem exclusivo quando a mensagem for colocada e o colocará no descritor de mensagens enviado com a mensagem. O gerenciador de filas também retorna esse identificador de mensagens no descritor de mensagem pertencente ao aplicativo de envio. O aplicativo pode usar esse valor para registrar informações sobre mensagens específicas e responder a consultas de outras partes do aplicativo.

Um MDMID gerado pelo gerenciador de filas consiste em um identificador de produto de 4 bytes (AMQ→ ou CSQ→ em ASCII ou EBCDIC, em que → representa um único caractere em branco), seguido por uma implementação específica do produto de uma sequência exclusiva. Em IBM MQ, isso

contém os primeiros 12 caracteres do nome do gerenciador de fila e um valor derivado do relógio do sistema.. Todos os gerenciadores de filas que podem se intercomunicar devem, portanto, ter nomes diferentes nos primeiros 12 caracteres, para assegurar que os identificadores de mensagens sejam exclusivos. A capacidade de gerar uma sequência exclusiva também depende do relógio do sistema não ser alterado para trás.. Para eliminar a possibilidade de um identificador de mensagem gerado pelo gerenciador de filas duplicando um gerado pelo aplicativo, o aplicativo deve evitar gerar identificadores com caracteres iniciais no intervalo de A a I em ASCII ou EBCDIC (X'41 'a X'49' e X'C1'a X'C9'). Entretanto, o aplicativo não é impedido de gerar identificadores com caracteres iniciais nesses intervalos.

Se a mensagem estiver sendo colocada em um tópico, o gerenciador de filas gerará identificadores de mensagens exclusivos conforme necessário para cada mensagem publicada. Se PMNMID for especificado pelo aplicativo, o gerenciador de fila gera um identificador de mensagem exclusivo para retornar na saída. Se MINONE for especificado pelo aplicativo, o valor do campo MDMID no MQMD permanecerá inalterado no retorno da chamada.

Consulte a descrição de PMRET em [PMOPT](#) para obter mais detalhes sobre as publicações retidas

Se a mensagem estiver sendo colocada em uma lista de distribuição, o gerenciador de filas gerará identificadores exclusivos de mensagens conforme necessário, mas o valor do campo MDMID no MQMD não será alterado no retorno da chamada, mesmo se MINONE ou PMNMID foi especificado. Se o aplicativo precisar saber os identificadores de mensagens gerados pelo gerenciador de filas, o aplicativo deverá fornecer registros MQPMR que contenham o campo PRMID

O aplicativo de envio também pode especificar um valor específico para o identificador de mensagem, diferente de MINONE; isso para o gerenciador de fila gerando um identificador de mensagem exclusivo.. Um aplicativo que está encaminhando uma mensagem pode usar esse recurso para propagar o identificador da mensagem original.

O próprio gerenciador de filas não faz uso desse campo, exceto para:

- Gerar um valor exclusivo, se solicitado, conforme descrito anteriormente
- Entregar o valor para o aplicativo que emite a solicitação get para a mensagem
- Copie o valor para o campo MDCID de qualquer mensagem de relatório que ele gera sobre essa mensagem (dependendo das opções MDREP )

Quando o gerenciador de filas ou um agente do canal de mensagens gera uma mensagem de relatório, ele configura o campo MDMID na maneira especificada pelo campo MDREP da mensagem original, RONMI ou ROPMI... Os aplicativos que geram mensagens de relatório também devem fazer isso:

Para a chamada MQGET, MDMID é um dos cinco campos que podem ser usados para selecionar uma mensagem específica a ser recuperada da fila. Normalmente, a chamada MQGET retorna a próxima mensagem na fila, mas se uma mensagem específica for necessária, isso pode ser obtido especificando um ou mais dos cinco critérios de seleção, em qualquer combinação; esses campos são:

- MDMID
- MDCID
- MDGID
- MDSEQ
- MDOFF

O aplicativo configura um ou mais desses campos para os valores necessários e, em seguida, configura as opções correspondentes de MO\* no campo GMMO em MQGMO para indicar que esses campos devem ser usados como critérios de seleção. Apenas as mensagens que possuem os valores especificados nesses campos são candidatas para recuperação O padrão para o campo GMMO (se não alterado pelo aplicativo) é corresponder ao identificador de mensagem e ao identificador de correlação.

Normalmente, a mensagem retornada é a primeira mensagem na fila que satisfaz os critérios de seleção. Mas se GMBRWN for especificado, a mensagem retornada será a próxima mensagem que satisfaz os critérios de seleção; a varredura para esta mensagem começa com a mensagem seguindo a posição atual do cursor.

**Nota:** A fila é varrida sequencialmente para uma mensagem que satisfaça os critérios de seleção, portanto, os tempos de recuperação serão mais lentos do que se nenhum critério de seleção for especificado, especialmente se muitas mensagens precisarem ser varridas antes que uma adequada seja localizada.

Consulte [Tabela 1](#) para obter mais informações sobre como os critérios de seleção são usados em várias situações.

Especificar MINONE como identificador de mensagem tem o mesmo efeito que não especificar MOMSGI, ou seja, qualquer identificador de mensagem será correspondente.

Esse campo será ignorado se a opção GMMUC for especificada no parâmetro **GMO** na chamada MQGET

No retorno de uma chamada MQGET, o campo MDMID é configurado para o identificador de mensagem da mensagem retornada (se houver).

O valor especial a seguir pode ser usado:

#### **MINONE**

Nenhum identificador de mensagem foi especificado.

O valor é binário zero para o comprimento do campo.

Este é um campo de entrada / saída para as chamadas MQGET, MQPUT e MQPUT1 . O comprimento deste campo é fornecido por LNMID O valor inicial deste campo é MINONE.

### **MDMT (número inteiro assinado de 10 dígitos)**

Tipo de mensagem.

Isso indica o tipo da mensagem. Os tipos de mensagens são agrupados da seguinte forma:

#### **MTSFST**

Valor mais baixo para tipos de mensagens definidas pelo sistema.

#### **MTSLST**

Valor mais alto para tipos de mensagens definidos pelo sistema.

Os valores a seguir estão definidos atualmente no intervalo do sistema:

#### **MTDGRM**

A mensagem não requer uma resposta

A mensagem é uma que não requer resposta.

#### **MTRQST**

Mensagem que requer uma resposta

A mensagem é aquela que requer uma resposta.

O nome da fila para a qual a resposta deve ser enviada deve ser especificado no campo MDRQ O campo MDREP indica como MDMID e MDCID da resposta devem ser configurados.

#### **MTRPLY**

Responda a uma mensagem de solicitação anterior

A mensagem é a resposta a uma mensagem de pedido anterior (MTRQST) A mensagem deve ser enviada para a fila indicada pelo campo MDRQ da mensagem de solicitação O campo MDREP da solicitação deve ser usado para controlar como MDMID e MDCID da resposta são configurados.

**Nota:** O gerenciador de filas não impinge o relacionamento de solicitação-resposta; esta é uma responsabilidade do aplicativo

#### **MTRPRT**

Mensagem de relatório

A mensagem está relatando alguma ocorrência esperada ou inesperada, geralmente relacionada a alguma outra mensagem (por exemplo, foi recebida uma mensagem de solicitação que continha dados inválidos). A mensagem deve ser enviada para a fila indicada pelo campo MDRQ do descritor de mensagens da mensagem original. O campo MDFB deve ser configurado para indicar a natureza do relatório.. O campo MDREP da mensagem original pode ser usado para controlar como MDMID e MDCID da mensagem de relatório devem ser configurados.

As mensagens de relatório geradas pelo gerenciador de filas ou agente do canal de mensagem são sempre enviadas para a fila MDRQ , com os campos MDFB e MDCID configurados conforme descrito anteriormente.

Outros valores dentro do intervalo do sistema podem ser definidos em versões futuras do MQI e são aceitos pelas chamadas MQPUT e MQPUT1 sem erro.

Os valores definidos pelo aplicativo também podem ser usados. Eles devem estar dentro do seguinte intervalo:

#### **MTAFST**

Valor mais baixo para tipos de mensagens definidas pelo aplicativo.

#### **MTALST**

Valor mais alto para tipos de mensagens definidas pelo aplicativo.

Para as chamadas MQPUT e MQPUT1 , o valor MDMT deve estar dentro do intervalo definido pelo sistema ou do intervalo definido pelo aplicativo; se não estiver, a chamada falhará com o código de razão RC2029.

Esse é um campo de saída para a chamada MQGET e um campo de entrada para as chamadas MQPUT e MQPUT1 . O valor inicial deste campo é MTDGRM..

### **MDOFF (número inteiro assinado de 10 dígitos)**

Deslocamento dos dados na mensagem física a partir do início da mensagem lógica.

Este é o deslocamento em bytes dos dados na mensagem física a partir do início da mensagem lógica da qual os dados fazem parte. Esses dados são chamados *segmento*. O deslocamento está no intervalo de 0 a 999 999 999. Uma mensagem física que não é um segmento de uma mensagem lógica tem um deslocamento zero.

Este campo não precisa ser configurado pelo aplicativo na chamada MQPUT ou MQGET se:

- Na chamada MQPUT, PMLOGO é especificado..
- Na chamada MQGET, o MOOFFS não é especificado

Estas são as maneiras recomendadas de usar estas chamadas para mensagens que não são mensagens de relatório. No entanto, se o aplicativo não estiver em conformidade com essas condições ou a chamada for MQPUT1, o aplicativo deverá assegurar que MDOFF seja configurado para um valor apropriado.

Na entrada para as chamadas MQPUT e MQPUT1 , o gerenciador de filas usa o valor detalhado na [Tabela 1](#). Na saída das chamadas MQPUT e MQPUT1 , o gerenciador de filas configura esse campo para o valor que foi enviado com a mensagem..

Para um relatório de mensagem de relatório em um segmento de uma mensagem lógica, o campo MDOLN (desde que não seja OLUNDF) é usado para atualizar o deslocamento nas informações do segmento retidas pelo gerenciador de filas.

Na entrada para a chamada MQGET, o gerenciador de filas usa o valor detalhado na [Tabela 1](#). Na saída da chamada MQGET, o gerenciador de filas configura esse campo para o valor da mensagem recuperada.

O valor inicial deste campo é zero. Esse campo será ignorado se MDVER for menor que MDVER2.

### **MDOLN (número inteiro assinado de 10 dígitos)**

Comprimento da mensagem original.

Esse campo é relevante apenas para mensagens de relatório que são segmentos. Especifica o comprimento do segmento da mensagem ao qual a mensagem de relatório se relaciona; não especifica o comprimento da mensagem lógica da qual o segmento faz parte, nem o comprimento dos dados na mensagem de relatório.

**Nota:** Ao gerar uma mensagem de relatório para uma mensagem que é um segmento, o gerenciador de filas e o agente do canal de mensagens copiam no MQMD para a mensagem de relatório os campos MDGID, MDSEQ, MDOFFe *MDMFL* da mensagem original. Como resultado, a mensagem de relatório também é um segmento. Aplicativos que geram mensagens de relatório são recomendados para fazer o mesmo e assegurar que o campo MDOLN esteja configurado corretamente.

O seguinte valor especial é definido:

#### **OLUNDF**

Comprimento original da mensagem não definido.

MDOLN é um campo de entrada nas chamadas MQPUT e MQPUT1, mas o valor fornecido pelo aplicativo é aceito apenas em circunstâncias específicas:

- Se a mensagem que está sendo colocada for um segmento e também uma mensagem de relatório, o gerenciador de filas aceita o valor especificado. O valor deve ser:
  - Maior que zero se o segmento não for o último segmento
  - Não menor que zero se o segmento for o último segmento
  - Não inferior ao comprimento dos dados presentes na mensagem

Se essas condições não forem satisfeitas, a chamada falhará com o código de razão RC2252.

- Se a mensagem que está sendo colocada for um segmento, mas não uma mensagem de relatório, o gerenciador de filas ignorará o campo e usará o comprimento dos dados da mensagem do aplicativo.
- Em todos os outros casos, o gerenciador de filas ignora o campo e usa o valor OLUNDF no lugar.

Este é um campo de saída na chamada MQGET

O valor inicial deste campo é OLUNDF. Esse campo será ignorado se MDVER for menor que MDVER2.


#### **MDPAN (sequência de caracteres de 28 bytes)**

Nome do aplicativo que insere a mensagem.

Isso faz parte do *contexto de origem* da mensagem.. Para obter mais informações sobre o contexto de mensagem, consulte [Contexto da mensagem](#) e [Informações de contexto de controle](#).

O formato do MDPAN depende do valor de MDPAT.

Quando esse campo for configurado pelo gerenciador de filas (ou seja, para todas as opções, exceto PMSETA), ele será configurado para o valor determinado pelo ambiente:

-  No z/OS, o gerenciador de filas usa:
  - Para o lote z/OS, o nome da tarefa de 8 caracteres do cartão JES JOB
  - Para TSO, o identificador de usuário do TSO de 7 caracteres
  - Para CICS, o applid de 8 caracteres, seguido pelo tranid de 4 caracteres
  - Para IMS, o identificador do sistema IMS de 8 caracteres, seguido pelo nome do PSB de 8 caracteres
  - Para XCF, o nome do grupo XCF de 8 caracteres, seguido pelo nome do membro XCF de 16 caracteres
  - Para uma mensagem gerada por um gerenciador de filas, os primeiros 28 caracteres do nome do gerenciador de filas
  - Para o enfileiramento distribuído sem CICS, o nome da tarefa de 8 caracteres do inicializador de canais seguido pelo nome de 8 caracteres do módulo colocado na fila de mensagens não entregues seguido por um identificador de tarefa de 8 caracteres.



- Para o processamento de ligações de linguagem do MQSeries Java com IBM MQ for z/OS o nome da tarefa de 8 caracteres do espaço de endereço criado para o ambiente z/OS UNIX System Services . Geralmente, esse será um identificador de usuário do TSO com um único caractere numérico anexado

Cada nome ou nomes são preenchidos à direita com espaços em branco, como qualquer espaço no restante do campo. Quando há mais de um nome, não há separador entre eles

- **Windows** Nos sistemas PC DOS e Windows , o gerenciador de filas usa:
  - Para um aplicativo CICS , o nome da transação CICS
  - Para um aplicativo nãoCICS , os 28 caracteres mais à direita do nome completo do executável
- **IBM i** No IBM i, o gerenciador de filas usa o nome completo da tarefa..
- **Linux** **AIX** No AIX and Linux, o gerenciador de filas usa:
  - Para um aplicativo CICS , o nome da transação CICS
  - Para um aplicativo nãoCICS , os 14 caracteres mais à direita do nome completo do executável se isso estiver disponível para o gerenciador de filas e, caso contrário, espaços em branco (por exemplo, em AIX)
- Em VSE/ESA, o gerenciador de fila usa o applid de 8 caracteres, seguido pelo tranid de 4 caracteres

Para as chamadas MQPUT e MQPUT1 , este é um campo de entrada /saída se PMSETA for especificado no parâmetro **PMO** . Quaisquer informações após um caractere nulo no campo são descartadas. O caractere nulo e quaisquer caracteres a seguir são convertidos em branco pelo gerenciador de filas. Se o PMSETA não for especificado, esse campo será ignorado na entrada e será um campo somente de saída

Este é um campo de saída para a chamada MQGET O comprimento deste campo é fornecido por LNPAN O valor inicial deste campo é de 28 caracteres em branco.

### **MDPAT (número inteiro assinado de 10 dígitos)**

Tipo de aplicativo que coloca a mensagem.

Isso faz parte do **contexto de origem** da mensagem.. Para obter mais informações sobre o contexto de mensagem, consulte [Contexto da mensagem](#) e [Informações de contexto de controle](#).

*MDPAT* pode ter um dos seguintes tipos padrão. Tipos definidos pelo usuário também podem ser usados, mas devem ser restritos a valores no intervalo ATUFST a ATULST.

#### **ATAIX**

AIX (mesmo valor que ATUNIX).

#### **ATBRKR**

Broker.

#### **ATCICS**

Transação CICS .

#### **ATCICB**

CICS bridge.

#### **ATVSE**

Transação CICS/VSE .

#### **ATDOS**

IBM MQ MQI client aplicação no PC DOS.

#### **ATDQM**

Agente do gerenciador de filas distribuído..

#### **ATGUAR**

Aplicativo Guardian Tandem (mesmo valor que ATNSK)

#### **ATIMS**

IMS .

**ATIMSB**

Ponte do IMS

**ATJAVA**

Java.

**ATMVS**

Aplicativo MVS ou TSO (mesmo valor que ATZOS).

**ATNOTE**

Lotus Notes Aplicativo do agente..

**ATNSK**

Tandem NonStop Aplicativo Kernel.

**AT390**

OS/390 (mesmo valor que ATZOS).

**AT400**

IBM i .

**ATQM**

Gerenciador de Filas

**ATUNIX**

UNIX .

**ATVOS**

Aplicativo do Stratus VOS

**ATWIN**

Aplicativo Windows de 16 bits

**ATWINT**

aplicativo Windows de 32 bits.

**ATXCF**

XCF.

**ATZOS**

z/OS .

**ATDEF**

Tipo de aplicativo padrão..

Este é o tipo de aplicativo padrão para a plataforma na qual o aplicativo está em execução

**Nota:** O valor dessa constante é específico para o ambiente.

**ATUNK**

Tipo de aplicativo desconhecido.

Esse valor pode ser usado para indicar que o tipo de aplicativo é desconhecido, embora outras informações de contexto estejam presentes.

**ATUFST**

Valor mais baixo para o tipo de aplicativo definido pelo usuário.

**ATULST**

Valor mais alto para o tipo de aplicativo definido pelo usuário.

O seguinte valor especial também pode ocorrer:

**ATNCON**

Não há informações de contexto presentes na mensagem

Este valor é configurado pelo gerenciador de filas quando uma mensagem é colocada sem contexto (ou seja, a opção de contexto PMNOC é especificada)...


Quando uma mensagem é recuperada, o MDPAT pode ser testado para esse valor para decidir se a mensagem tem contexto (é recomendado que o MDPAT nunca seja configurado para ATNCON,

por um aplicativo usando PMSETA, se qualquer um dos outros campos de contexto não estiver em branco)

### **ATSIB**

Indica uma mensagem originada em outro produto de sistema de mensagens do IBM MQ e que chegou por meio da ponte SIB (Service Integration Bus)

Quando o gerenciador de filas gera essas informações como resultado de uma colocação de aplicativo, o campo é configurado para um valor determinado pelo ambiente.

 Observe que no IBM i, o campo é configurado como AT400; o gerenciador de filas nunca usa o ATCICS no IBM i

Para as chamadas MQPUT e MQPUT1, este é um campo de entrada /saída se PMSETA for especificado no parâmetro **PMO**. Se o PMSETA não for especificado, esse campo será ignorado na entrada e será um campo somente de saída

Após a conclusão bem-sucedida de uma chamada MQPUT ou MQPUT1, esse campo contém o MDPAT que foi transmitido com a mensagem se ele foi colocado em uma fila. Esse será o valor de MDPAT que será mantido com a mensagem se ela for retida (consulte a descrição de PMRET para obter mais detalhes sobre as publicações retidas), mas não será usado como o MDPAT quando a mensagem for enviada como uma publicação para os assinantes, pois eles fornecem um valor para substituir MDPAT em todas as publicações enviadas para eles. Se a mensagem não tiver nenhum contexto, o campo será configurado como ATNCON

Este é um campo de saída para a chamada MQGET O valor inicial deste campo é ATNCON.

### **MDPD (cadeia de caracteres de 8 bytes).**

Data quando a mensagem foi colocada.

Isso faz parte do *contexto de origem* da mensagem.. Para obter mais informações sobre o contexto de mensagem, consulte [Contexto da mensagem](#) e [Informações de contexto de controle](#).

O formato utilizado para a data em que esse campo é gerado pelo gerenciador de filas é:

- AAAAMMDD

em que os caracteres representam:

#### **AAAA**

ano (quatro dígitos numéricos)

#### **MM**

mês do ano (01 a 12)

#### **DD**

dia do mês (01 a 31)

O Tempo Médio de Greenwich (GMT) é usado para os campos MDPD e MDPT, sujeito ao relógio do sistema sendo configurado com precisão para GMT.

Se a mensagem foi colocada como parte de uma unidade de trabalho, a data é quando a mensagem foi colocada e não a data em que a unidade de trabalho foi confirmada.

Para as chamadas MQPUT e MQPUT1, este é um campo de entrada /saída se PMSETA for especificado no parâmetro **PMO**. O conteúdo do campo não é verificado pelo gerenciador de filas, exceto que qualquer informação após um caractere nulo no campo é descartada. O caractere nulo e quaisquer caracteres a seguir são convertidos em branco pelo gerenciador de filas. Se o PMSETA não for especificado, esse campo será ignorado na entrada e será um campo somente de saída

Após a conclusão bem-sucedida de uma chamada MQPUT ou MQPUT1, esse campo contém o MDPD que foi transmitido com a mensagem se ele foi colocado em uma fila. Esse será o valor de MDPD que será mantido com a mensagem se ela for retida (consulte a descrição de PMRET para obter mais detalhes sobre as publicações retidas), mas não será usado como o MDPD quando a mensagem for enviada como uma publicação para os assinantes, pois eles fornecem um valor para substituir MDPD

em todas as publicações enviadas para eles. Se a mensagem não tiver nenhum contexto, o campo estará inteiramente em branco

Este é um campo de saída para a chamada MQGET O comprimento deste campo é fornecido por LNPDAT O valor inicial deste campo é 8 caracteres em branco.

### **MDPER (número inteiro assinado de 10 dígitos)**

Persistência da mensagem.

Isso indica se a mensagem sobrevive a falhas e reinicializações do sistema do gerenciador de filas Para as chamadas MQPUT e MQPUT1 , o valor deve ser um dos seguintes:

#### **PEPER**

A mensagem é persistente

Isso significa que a mensagem sobrevive a falhas e reinicializações do sistema do gerenciador de fila Quando a mensagem tiver sido colocada e a unidade de trabalho do putter confirmada (se a mensagem for colocada como parte de uma unidade de trabalho), a mensagem será preservada no armazenamento auxiliar. Ela permanece lá até que a mensagem seja removida da fila e a unidade de trabalho do getter confirmada (se a mensagem for recuperada como parte de uma unidade de trabalho).

Quando uma mensagem persistente é enviada para uma fila remota, um mecanismo de armazenamento e encaminhamento é usado para manter a mensagem em cada gerenciador de filas ao longo da rota para o destino, até que a mensagem seja conhecida por ter chegado ao próximo gerenciador de filas

Mensagens persistentes não podem ser colocadas em:

- Filas dinâmicas temporárias
- Filas compartilhadas em que o nível de estrutura do recurso de acoplamento é menor que três ou a estrutura do recurso de acoplamento não é recuperável.

As mensagens persistentes podem ser colocadas em filas dinâmicas permanentes, filas predefinidas e filas compartilhadas onde o nível da estrutura do recurso de acoplamento é 3 e o recurso de acoplamento é recuperável.

#### **PENPER**

A mensagem não é persistente

Isso significa que a mensagem normalmente não sobrevive a falhas ou reinicializações do sistema do gerenciador de fila Isso se aplicará mesmo se uma cópia intacta da mensagem for localizada no armazenamento auxiliar durante a reinicialização do gerenciador de filas

No caso especial de filas compartilhadas, as mensagens não persistentes *não* sobrevivem a reinicializações de gerenciadores de fila no grupo de filas compartilhadas, mas não sobrevivem a falhas do recurso de acoplamento usado para armazenar mensagens nas filas compartilhadas

#### **PEQDEF**

Mensagem tem persistência padrão.

- Se a fila for uma fila de clusters, a persistência da mensagem será obtida do atributo **DefPersistence** definido no gerenciador de filas de destino que possui a instância específica da fila na qual a mensagem é colocada. Geralmente, todas as instâncias de uma fila de clusters têm o mesmo valor para o atributo **DefPersistence** , embora isso não seja obrigatório..

O valor de **DefPersistence** é copiado no campo *MDPER* quando a mensagem é colocada na fila de destino. Se **DefPersistence** for alterado subsequentemente, as mensagens que já foram colocadas na fila não serão afetadas

- Se a fila não for uma fila de clusters, a persistência da mensagem será obtida do atributo **DefPersistence** definido no gerenciador de fila local, mesmo se o gerenciador de fila de destino for remoto.

Se houver mais de uma definição no caminho de resolução do nome da fila, a persistência padrão será obtida do valor desse atributo na primeira definição no caminho. Isto poderia ser:

- uma fila de alias
- Uma fila local
- Uma definição local de uma fila remota
- Um alias do gerenciador de filas
- Uma fila de transmissão (por exemplo, a fila DefXmitQName )

O valor de **DefPersistence** é copiado no campo MDPER quando a mensagem é colocada. Se **DefPersistence** for alterado subsequentemente, as mensagens que já foram colocadas não serão afetadas

As mensagens persistentes e não persistentes podem existir na mesma fila.

Ao responder a uma mensagem, os aplicativos devem normalmente usar para a mensagem de resposta a persistência da mensagem de solicitação.

Para uma chamada MQGET, o valor retornado é PEPER ou PENPER.

Esse é um campo de saída para a chamada MQGET e um campo de entrada para as chamadas MQPUT e MQPUT1 . O valor inicial desse campo é PEQDEF.

### **MDPRI (número inteiro assinado de 10 dígitos)**

Prioridade da mensagem.

Para as chamadas MQPUT e MQPUT1 , o valor deve ser maior ou igual a zero; zero é a prioridade mais baixa. O seguinte valor especial também pode ser usado:

#### **PRQDEF**

Prioridade padrão para fila.

- Se a fila for uma fila de clusters, a prioridade para a mensagem será obtida do atributo **DefPriority** , conforme definido no gerenciador de filas de destino que possui a instância específica da fila na qual a mensagem é colocada. Geralmente, todas as instâncias de uma fila de clusters têm o mesmo valor para o atributo **DefPriority** , embora isso não seja obrigatório..

O valor de **DefPriority** é copiado no campo MDPRI quando a mensagem é colocada na fila de destino. Se **DefPriority** for alterado subsequentemente, as mensagens que já foram colocadas na fila não serão afetadas

- Se a fila não for uma fila de clusters, a prioridade para a mensagem será obtida do atributo **DefPriority** , conforme definido no gerenciador de fila local, mesmo se o gerenciador de filas de destino for remoto.

Se houver mais de uma definição no caminho de resolução de nome da fila, a prioridade padrão será obtida do valor desse atributo na primeira definição no caminho. Isto poderia ser:

- uma fila de alias
- Uma fila local
- Uma definição local de uma fila remota
- Um alias do gerenciador de filas
- Uma fila de transmissão (por exemplo, a fila DefXmitQName )

O valor de **DefPriority** é copiado no campo MDPRI quando a mensagem é colocada. Se **DefPriority** for alterado subsequentemente, as mensagens que já foram colocadas não serão afetadas

O valor retornado pela chamada MQGET é sempre maior ou igual a zero; o valor PRQDEF nunca é retornado.

Se uma mensagem for colocada com uma prioridade maior que o máximo suportado pelo gerenciador de filas locais (esse máximo é fornecido pelo atributo do gerenciador de fila **MaxPriority**), a mensagem será aceita pelo gerenciador de filas, mas será colocada na fila na prioridade máxima do gerenciador de filas; a chamada MQPUT ou MQPUT1 será concluída com CCWARN e código de razão RC2049. Entretanto, o campo MDPRI retém o valor especificado pelo aplicativo que colocou a mensagem.

Ao responder a uma mensagem, os aplicativos devem normalmente usar para a mensagem de resposta a prioridade da mensagem de solicitação. Em outras situações, especificar PRQDEF permite que o ajuste de prioridade seja executado sem alterar o aplicativo.

Esse é um campo de saída para a chamada MQGET e um campo de entrada para as chamadas MQPUT e MQPUT1. O valor inicial desse campo é PRQDEF.

### **MDPT (cadeia de caracteres de 8 bytes)**

Hora quando a mensagem foi colocada.

Isso faz parte do **contexto de origem** da mensagem.. Para obter mais informações sobre o contexto de mensagem, consulte [Contexto da mensagem](#) e [Informações de contexto de controle](#).

O formato usado para o horário em que esse campo é gerado pelo gerenciador de filas é:

- HHMMSSTH

em que os caracteres representam (por ordem):

#### **DOMS**

horas (00 a 23)

#### **MM**

minutos (00 a 59)

#### **SS**

segundos (00 a 59; consulte [nota](#))

#### **T**

Décimos de segundo (0 a 9)

#### **H**

centésimos de segundo (0 a 9)

**Nota:** Se o relógio do sistema for sincronizado com um padrão de tempo muito preciso, será possível em raras ocasiões que 60 ou 61 sejam retornados para os segundos em MDPT. Isso acontece quando os segundos bissextos são inseridos no padrão de tempo global.

O Tempo Médio de Greenwich (GMT) é usado para os campos MDPD e MDPT, sujeito ao relógio do sistema sendo configurado com precisão para GMT.

Se a mensagem foi colocada como parte de uma unidade de trabalho, o horário será quando a mensagem foi colocada e não o horário em que a unidade de trabalho foi confirmada.

Para as chamadas MQPUT e MQPUT1, este é um campo de entrada /saída se PMSETA for especificado no parâmetro **PMO**. O conteúdo do campo não é verificado pelo gerenciador de filas, exceto que qualquer informação após um caractere nulo no campo é descartada. O caractere nulo e quaisquer caracteres a seguir são convertidos em branco pelo gerenciador de filas. Se o PMSETA não for especificado, esse campo será ignorado na entrada e será um campo somente de saída.

Após a conclusão bem-sucedida de uma chamada MQPUT ou MQPUT1, esse campo contém o valor MDPT que foi transmitido com a mensagem se ele foi colocado em uma fila. Esse será o valor de MDPT que será mantido com a mensagem se ela for retida (consulte a descrição de PMRET para obter mais detalhes sobre as publicações retidas), mas não será usado como o MDPT quando a mensagem for enviada como uma publicação para os assinantes, pois eles fornecem um valor para substituir MDPT em todas as publicações enviadas para eles. Se a mensagem não tiver nenhum contexto, o campo estará inteiramente em branco.

Este é um campo de saída para a chamada MQGET. O comprimento deste campo é fornecido por LNPTIM. O valor inicial deste campo é 8 caracteres em branco.

## MDREP (número inteiro assinado de 10 dígitos)

Opções para mensagens de relatório.

Uma mensagem de relatório é uma mensagem sobre outra mensagem, usada para informar um aplicativo sobre eventos esperados ou inesperados relacionados à mensagem original. O campo MDREP permite que o aplicativo que está enviando a mensagem original especifique quais mensagens de relatório são necessárias, se os dados da mensagem do aplicativo devem ser incluídos neles e também (para relatórios e respostas) como a mensagem e os identificadores de correlação no relatório ou mensagem de resposta devem ser configurados. Qualquer ou todos (ou nenhum) dos seguintes tipos de mensagem de relatório pode ser solicitado:

- Exceção
- Expiração
- Confirmar na chegada (COA)
- Confirmar na entrega (COD)
- notificação de ação positiva (PAN)
- notificação de ação negativa (NAN)

Se mais de um tipo de mensagem de relatório for necessário ou se outras opções de relatório forem necessárias, os valores poderão ser incluídos juntos (não inclua a mesma constante mais de uma vez)

O aplicativo que recebe a mensagem de relatório pode determinar o motivo pelo qual o relatório foi gerado examinando o campo MDFB no MQMD; consulte o campo MDFB para obter mais detalhes..

O uso de opções de relatório ao colocar uma mensagem em um tópico pode fazer com que zero, uma ou várias mensagens de relatório sejam geradas e enviadas para o aplicativo. Isto porque a mensagem de publicação pode ser enviada para zero, um ou muitos aplicativos de assinatura.

**Opções de exceção:** É possível especificar uma das seguintes opções para solicitar uma mensagem de relatório de exceção.

## ROACTIVITY

Relatórios de atividades necessários

Essa opção de relatório permite que um relatório de atividades seja gerado, sempre que uma mensagem com esse conjunto de opções de relatório for processada por aplicativos de suporte

As mensagens com essa opção de relatório configurada devem ser aceitas por qualquer gerenciador de filas, mesmo se elas não 'entenderem' a opção. Isso permite que a opção de relatório seja configurada em qualquer mensagem do usuário, mesmo se ela for processada por gerenciadores de filas anteriores Para conseguir isso, a opção de relatório é colocada no subcampo ROAUM.

Se um processo (um gerenciador de filas ou um processo do usuário) executar uma atividade em uma mensagem com ROACT configurado, ele poderá optar por gerar e colocar um relatório de atividades.

A opção de relatório de atividades permite que a rota de qualquer mensagem seja rastreada em toda uma rede do gerenciador de filas A opção de relatório pode ser especificada em qualquer mensagem do usuário atual e instantaneamente eles podem começar a calcular a rota da mensagem através da rede. Se o aplicativo que está gerando a mensagem não puder ativar a geração de relatórios de atividades, ele poderá ser ativado usando uma saída cruzada da API fornecida pelos administradores do gerenciador de filas

Várias condições são aplicáveis aos relatórios de atividades:

1. A rota será menos detalhada se houver menos gerenciadores de filas na rede que possam gerar relatórios de atividades.
2. Os relatórios de atividades podem não ser facilmente "ordenáveis" para determinar a rota tomada.
3. Os relatórios de atividades podem não conseguir localizar uma rota para seu destino solicitado.

## ROEXC

Relatórios de exceção necessários.

Esse tipo de relatório pode ser gerado por um agente do canal de mensagens quando uma mensagem é enviada para outro gerenciador de fila e a mensagem não pode ser entregue para a fila de destino especificada... Por exemplo, a fila de destino ou uma fila de transmissão intermediária pode estar cheia, ou a mensagem pode ser muito grande para a fila

A geração da mensagem de relatório de exceção depende da persistência da mensagem original e da velocidade do canal de mensagens (normal ou rápido) por meio do qual a mensagem original viaja:

- Para todas as mensagens persistentes e para mensagens não persistentes que viajam por canais de mensagens normais, o relatório de exceção será gerado somente se a ação especificada pelo aplicativo de envio para a condição de erro puder ser concluída com êxito. O aplicativo de envio pode especificar uma das seguintes ações para controlar a disposição da mensagem original quando surgir a condição de erro:

- RODLQ (isso faz com que a mensagem original seja colocada na fila de mensagens não entregues)
- RODISC (isso faz com que a mensagem original seja descartada)

Se a ação especificada pelo aplicativo de envio não puder ser concluída com êxito, a mensagem original será deixada na fila de transmissão e nenhuma mensagem de relatório de exceção será gerada..

- Para mensagens não persistentes que viajam por canais de mensagens rápidos, a mensagem original é removida da fila de transmissão e o relatório de exceção gerado, mesmo se a ação especificada para a condição de erro não puder ser concluída com êxito. Por exemplo, se RODLQ for especificado, mas a mensagem original não puder ser colocada na fila de mensagens não entregues porque (digamos) essa fila está cheia, a mensagem do relatório de exceções será gerada e a mensagem original descartada.

Consulte [Persistência de Mensagem](#) para obter mais informações sobre canais de mensagem normais e rápidos

Um relatório de exceção não será gerado se o aplicativo que colocou a mensagem original puder ser notificado de forma síncrona sobre o problema por meio do código de razão retornado pela chamada MQPUT ou MQPUT1 .

Os aplicativos também podem enviar relatórios de exceções, para indicar que uma mensagem recebida não pode ser processada (por exemplo, porque é uma transação de débito que faria com que a conta excedesse seu limite de crédito).

Os dados da mensagem original não são incluídos com a mensagem de relatório.

Não especifique mais de um ROEXC, ROEXCD e ROEXCF.

## ROEXCD

Relatórios de exceção com dados necessários.

Isso é o mesmo que ROEXC, exceto que os primeiros 100 bytes dos dados da mensagem do aplicativo da mensagem original são incluídos na mensagem de relatório. Se a mensagem original contiver uma ou mais estruturas de cabeçalho do MQ , elas serão incluídas na mensagem de relatório, além dos 100 bytes de dados do aplicativo

Não especifique mais de um ROEXC, ROEXCD e ROEXCF.

## ROEXCF

Relatórios de exceção com dados completos necessários.

Este é o mesmo que ROEXC, exceto que todos os dados da mensagem do aplicativo da mensagem original estão incluídos na mensagem de relatório

Não especifique mais de um ROEXC, ROEXCD e ROEXCF.



**Opções de expiração:** É possível especificar uma das opções a seguir para solicitar uma mensagem de relatório de expiração.

#### **ROEXP**

Relatórios de expiração necessários.

Esse tipo de relatório será gerado pelo gerenciador de filas se a mensagem for descartada antes da entrega para um aplicativo porque seu tempo de expiração passou (consulte o campo MDEXP). Se essa opção não for configurada, nenhuma mensagem de relatório será gerada se uma mensagem for descartada por essa razão (mesmo se uma das opções ROEXC\* for especificada).

Os dados da mensagem original não são incluídos com a mensagem de relatório.

Não especifique mais de um ROEXP, ROEXPD e ROEXPF.

#### **ROEXPD**

Relatórios de expiração com dados necessários.

Isso é o mesmo que ROEXP, exceto que os primeiros 100 bytes dos dados da mensagem do aplicativo da mensagem original estão incluídos na mensagem de relatório. Se a mensagem original contiver uma ou mais estruturas de cabeçalho do MQ, elas serão incluídas na mensagem de relatório, além dos 100 bytes de dados do aplicativo.

Não especifique mais de um ROEXP, ROEXPD e ROEXPF.

#### **ROEXPF**

Relatórios de expiração com dados completos necessários.

Isso é o mesmo que ROEXP, exceto que todos os dados da mensagem do aplicativo da mensagem original estão incluídos na mensagem de relatório.

Não especifique mais de um ROEXP, ROEXPD e ROEXPF.

**Confirmar opções de chegada:** é possível especificar uma das opções a seguir para solicitar uma mensagem de relatório de confirmação de chegada.

#### **ROCOA**

Relatórios de confirmação na chegada necessários.

Esse tipo de relatório é gerado por um gerenciador de filas que possui a fila de destino, quando a mensagem é colocada na fila de destino. Os dados da mensagem original não são incluídos com a mensagem de relatório.

Se a mensagem for colocada como parte de uma unidade de trabalho e a fila de destino for uma fila local, a mensagem de relatório COA gerada pelo gerenciador de filas se tornará disponível para recuperação somente se e quando a unidade de trabalho for confirmada.

Um relatório de COA não será gerado se o campo MDFMT no descritor de mensagens for FMXQH ou FMDLH. Isso evita que um relatório COA seja gerado se a mensagem for colocada em uma fila de transmissão ou não for entregue e colocada em uma fila de mensagens não entregues.

Não especifique mais de um ROCOA, ROCOAD e ROCOAF.

#### **ROCOAD**

Confirme os relatórios de chegada com os dados necessários.

É o mesmo que ROCOA, exceto que os primeiros 100 bytes dos dados da mensagem do aplicativo da mensagem original são incluídos na mensagem de relatório. Se a mensagem original contiver uma ou mais estruturas de cabeçalho do MQ, elas serão incluídas na mensagem de relatório, além dos 100 bytes de dados do aplicativo.

Não especifique mais de um ROCOA, ROCOAD e ROCOAF.

#### **ROCOAF**

Confirme os relatórios de chegada com os dados completos necessários.

Isso é o mesmo que ROCOA, exceto que todos os dados da mensagem do aplicativo da mensagem original estão incluídos na mensagem de relatório.

Não especifique mais de um ROCOA, ROCOAD e ROCOAF.

**Opções de descarte e expiração:** é possível especificar a opção a seguir para configurar o tempo de expiração e o sinalizador de descarte para mensagens de relatório.

#### **ROPDAE**

Configure o tempo de expiração da mensagem de relatório e descarte a sinalização.

Essa opção assegura que as mensagens de relatório e de resposta herdem o tempo de expiração e o sinalizador de descarte (descarte ou não) de suas mensagens originais. Com este conjunto de opções, mensagens de relatório e resposta:

1. Herde o sinalizador RODISC (se ele estiver configurado).
2. Herdar o tempo de expiração restante da mensagem, se a mensagem não for um relatório de expiração. Se a mensagem for um relatório de expiração, o tempo de expiração será configurado como 60 segundos

Com esse conjunto de opções, o seguinte se aplica:

#### **Nota:**

1. As mensagens de relatório e resposta são geradas com um sinalizador de descarte e um valor de expiração e não podem permanecer no sistema.
2. As mensagens de rota de rastreamento são impedidas de atingir filas de destino em gerenciadores de filas não ativados por rota de rastreamento..
3. As filas são impedidas de serem preenchidas com relatórios que não podem ser entregues, se links de comunicações forem interrompidos.
4. As respostas do servidor de comandos herdam o restante da expiração da solicitação

**Confirmar opções de entrega:** é possível especificar uma das opções a seguir para solicitar uma mensagem de relatório de confirmação de entrega.

#### **ROCOD**

Relatórios de confirmação na entrega necessários.

Esse tipo de relatório é gerado pelo gerenciador de filas quando um aplicativo recupera a mensagem da fila de destino de uma maneira que faz a mensagem ser excluída da fila. Os dados da mensagem original não são incluídos com a mensagem de relatório.

Se a mensagem for recuperada como parte de uma unidade de serviço, a mensagem de relatório será gerada dentro da mesma unidade de trabalho, de modo que o relatório não estará disponível até que a unidade de trabalho seja confirmada. Se a unidade de trabalho for restaurada, o relatório não será enviado..

Um relatório de COD não será gerado se o campo MDFMT no descritor de mensagens for FMDLH. Isso impede que um relatório COD seja gerado se a mensagem não puder ser entregue e colocada em uma fila de mensagens não entregues.

ROCOD não será válido se a fila de destino for uma fila XCF

Não especifique mais de um ROCOD, ROCODD e ROCODF.

#### **ROCODD**

Confirme os relatórios de entrega com os dados necessários

Este é o mesmo que ROCOD, exceto que os primeiros 100 bytes dos dados da mensagem do aplicativo da mensagem original são incluídos na mensagem de relatório. Se a mensagem original contiver uma ou mais estruturas de cabeçalho do MQ, elas serão incluídas na mensagem de relatório, além dos 100 bytes de dados do aplicativo

Se o GMATM for especificado na chamada MQGET para a mensagem original e a mensagem recuperada for truncada, a quantidade de dados da mensagem do aplicativo colocada na mensagem de relatório será o mínimo de:

- O comprimento da mensagem original

- 100 bytes.

ROCODD não será válido se a fila de destino for uma fila XCF

Não especifique mais de um ROCOD, ROCODD e ROCODF.

### **ROCODF**

Confirme relatórios de entrega com dados completos necessários.

Isso é o mesmo que ROCOD, exceto que todos os dados da mensagem do aplicativo da mensagem original estão incluídos na mensagem de relatório

ROCODF não será válido se a fila de destino for uma fila XCF

Não especifique mais de um ROCOD, ROCODD e ROCODF.

**Opções de notificação de ação:** é possível especificar uma ou ambas as opções a seguir para solicitar que o aplicativo de recebimento envie uma mensagem de relatório de ação positiva ou negativa.

### **ROPAN**

Relatórios de notificação de ação positiva necessários.

Esse tipo de relatório é gerado pelo aplicativo que recupera a mensagem e age sobre ela.. Ele indica que a ação solicitada na mensagem foi executada com êxito O aplicativo que gera o relatório determina se algum dado deve ser incluído com o relatório.

Além de transmitir essa solicitação para o aplicativo recuperando a mensagem, o gerenciador de filas não toma nenhuma ação com base nessa opção. É responsabilidade do aplicativo de recuperação gerar o relatório se apropriado.

### **RONAN**

Relatórios de notificação de ação negativa necessários..

Esse tipo de relatório é gerado pelo aplicativo que recupera a mensagem e age sobre ela.. Isso indica que a ação solicitada na mensagem não foi executada com êxito O aplicativo que gera o relatório determina se algum dado deve ser incluído com o relatório. Por exemplo, pode ser desejável incluir alguns dados indicando por que a solicitação não pôde ser executada.

Além de transmitir essa solicitação para o aplicativo recuperando a mensagem, o gerenciador de filas não toma nenhuma ação com base nessa opção. É responsabilidade do aplicativo de recuperação gerar o relatório se apropriado.

A determinação de quais condições correspondem a uma ação positiva e quais correspondem a uma ação negativa é da responsabilidade do pedido.. No entanto, recomenda-se que se a solicitação tiver sido executada apenas parcialmente, um relatório NAN em vez de um relatório PAN deve ser gerado se solicitado. Também é recomendado que cada condição possível deve corresponder a uma ação positiva, ou uma ação negativa, mas não ambas.

**Opções do identificador de mensagens:** é possível especificar uma das opções a seguir para controlar como o MDMID da mensagem de relatório (ou da mensagem de resposta) deve ser configurado.

### **RONMI**

Novo identificador de mensagem

Essa é a ação padrão e indica que se um relatório ou resposta for gerado como resultado dessa mensagem, um novo MDMID deverá ser gerado para a mensagem de relatório ou resposta.

### **ROPMI**

Identificador de mensagem de transmissão.

Se um relatório ou resposta for gerado como resultado dessa mensagem, o MDMID dessa mensagem será copiado para o MDMID da mensagem de relatório ou resposta.

O MsgId de uma mensagem de publicação será diferente para cada assinante que receber uma cópia da publicação e, portanto, o MsgId copiado na mensagem de relatório ou de resposta será diferente para cada um.

Se essa opção não for especificada, RONMI será assumido..

**Opções do identificador de correlação:** é possível especificar uma das opções a seguir para controlar como o MDCID da mensagem de relatório (ou da mensagem de resposta) deve ser configurado.

#### **ROCMTC**

Copie o identificador de mensagem para o identificador de correlação.

Essa é a ação padrão e indica que se um relatório ou resposta for gerado como resultado dessa mensagem, o MDMID dessa mensagem deverá ser copiado para o MDCID da mensagem de relatório ou resposta.

O MsgId de uma mensagem de publicação será diferente para cada assinante que recebe uma cópia da publicação e, portanto, o MsgId copiado no CorrelId da mensagem de relatório ou de resposta será diferente para cada um.

#### **ROPCI**

Identificador de correlação de transmissão

Se um relatório ou resposta for gerado como resultado dessa mensagem, o MDCID dessa mensagem será copiado para o MDCID da mensagem de relatório ou resposta.

O MDCID de uma mensagem de publicação será específico para um assinante, a menos que use a opção SOSCID e configure o campo SCDIC no MQSD para CINONE. Portanto, é possível que o MDCID copiado no MDCID da mensagem de relatório ou de resposta seja diferente para cada um.

Se esta opção não for especificada, ROCMTC será assumido.

Os servidores que responderem a solicitações ou gerarem mensagens de relatório são recomendados para verificar se as opções ROPMI ou ROPCI foram configuradas na mensagem original. Se eles foram, os servidores devem executar a ação descrita para essas opções. Se nenhum deles estiver configurado, os servidores deverão executar a ação padrão correspondente.

: É possível especificar uma das seguintes opções para controlar a disposição da mensagem original quando ela não puder ser entregue à fila de destino. Essas opções se aplicam apenas àquelas situações que resultariam em uma mensagem de relatório de exceção sendo gerada se uma tivesse sido solicitada pelo aplicativo de envio. O aplicativo pode configurar as opções de disposição independentemente de solicitar relatórios de exceção.

#### **RODLQ**

Coloque a mensagem na fila de mensagens não entregues.

Esta é a ação padrão e indica que a mensagem deve ser colocada na fila de mensagens não entregues, se a mensagem não puder ser entregue para a fila de destino. Isso acontece nas seguintes situações:

- Quando o aplicativo que colocou a mensagem original não pode ser notificado de forma síncrona sobre o problema por meio do código de razão retornado pela chamada MQPUT ou MQPUT1. Uma mensagem de relatório de exceção é gerada, se uma foi solicitada pelo emissor.
- Quando o aplicativo que colocou a mensagem original estava colocando em um tópico. Uma mensagem de relatório de exceção será gerada, se uma foi solicitada pelo emissor.

#### **RODISC**

Descarte a mensagem

Isso indica que a mensagem deve ser descartada se não puder ser entregue à fila de destino. Isso acontece nas seguintes situações:

- Quando o aplicativo que colocou a mensagem original não pode ser notificado de forma síncrona sobre o problema por meio do código de razão retornado pela chamada MQPUT ou MQPUT1. Uma mensagem de relatório de exceção é gerada, se uma foi solicitada pelo emissor.
- Quando o aplicativo que colocou a mensagem original estava colocando em um tópico. Uma mensagem de relatório de exceção será gerada, se uma foi solicitada pelo emissor.

Se for necessário retornar a mensagem original para o emissor, sem que a mensagem original seja colocada na fila de mensagens não entregues, o emissor deverá especificar RODISC com ROEXCF.

**Opção padrão:** será possível especificar o seguinte se nenhuma opção de relatório for necessária:..

#### **RONONE**

Nenhum relatório requerido.

Este valor pode ser utilizado para indicar que nenhuma outra opção foi especificada.. O RONONE é definido para auxiliar a documentação do programa Não se pretende que essa opção seja usada com qualquer outra, mas como seu valor é zero, tal uso não pode ser detectado.

#### **Informações gerais:**

1. Todos os tipos de relatório necessários devem ser solicitados especificamente pelo aplicativo enviando a mensagem original. Por exemplo, se um relatório COA for solicitado, mas um relatório de exceção não for, um relatório COA será gerado quando a mensagem for colocada na fila de destino, mas nenhum relatório de exceção será gerado se a fila de destino estiver cheia quando a mensagem chegar lá. Se nenhuma opção MDREP for configurada, nenhuma mensagem de relatório será gerada pelo gerenciador de filas ou agente do canal de mensagens (MCA).

Algumas opções de relatório podem ser especificadas, mesmo que o gerenciador de filas locais não as reconheça; isso é útil quando a opção deve ser processada pelo gerenciador de filas de destino Consulte [“Opções de relatório e sinalizações de mensagem no IBM i”](#) na página 1484 para obter mais detalhes.

Se uma mensagem de relatório for solicitada, o nome da fila para a qual o relatório deve ser enviado deverá ser especificado no campo MDRQ Quando uma mensagem de relatório é recebida, a natureza do relatório pode ser determinada examinando o campo MDFB no descritor de mensagens

2. Se o gerenciador de filas ou o MCA que gera uma mensagem de relatório não puder colocar a mensagem de relatório na fila de respostas (por exemplo, porque a fila de respostas ou a fila de transmissão está cheia), a mensagem de relatório será colocada na fila de mensagens não entregues. Se isso também falhar, ou não houver fila de devoluções, a ação tomada dependerá do tipo da mensagem de relatório:
  - Se a mensagem de relatório for um relatório de exceção, a mensagem que fez com que o relatório de exceção fosse gerado será deixada em sua fila de transmissão; isso assegura que a mensagem não seja perdida.
  - Para todos os outros tipos de relatório, a mensagem de relatório é descartada e o processamento continua normalmente Isso é feito porque a mensagem original já foi entregue com segurança (para mensagens de relatório COA ou COD) ou não é mais de interesse (para uma mensagem de relatório de expiração).

Quando uma mensagem de relatório for colocada com êxito em uma fila (a fila de destino ou uma fila de transmissão intermediária), a mensagem não estará mais sujeita a processamento especial; ela será tratada como qualquer outra mensagem.

3. Quando o relatório é gerado, a fila MDRQ é aberta e a mensagem de relatório é colocada usando a autoridade do MDUID no MQMD da mensagem que está causando o relatório, exceto nos seguintes casos:
  - Os relatórios de exceção gerados por um MCA de recebimento são colocados com qualquer autoridade que o MCA usou quando tentou colocar a mensagem que está causando o relatório. O atributo do canal CDPA determina o identificador de usuário utilizado..
  - Os relatórios COA gerados pelo gerenciador de fila são colocados com qualquer autoridade que foi usada quando a mensagem que causou o relatório foi colocada no gerenciador de filas que gerou o relatório. Por exemplo, se a mensagem foi colocada por um MCA de recebimento usando o identificador de usuário do MCA, o gerenciador de filas coloca o relatório COA usando o identificador de usuário do MCA.

Os aplicativos que geram relatórios devem normalmente usar a mesma autoridade que eles teriam usado para gerar uma resposta; normalmente, essa deve ser a autoridade do identificador de usuário na mensagem original

Se o relatório tiver que viajar para um destino remoto, os remetentes e receptores podem decidir se o aceitam, da mesma forma que o fazem para outras mensagens.

4. Se uma mensagem de relatório com dados for solicitada:
  - A mensagem de relatório é sempre gerada com a quantidade de dados solicitada pelo emissor da mensagem original. Se a mensagem de relatório for muito grande para a fila de respostas, o processamento descrito anteriormente ocorrerá; a mensagem de relatório nunca será truncada para se ajustar na fila de respostas
  - Se o MDFMT da mensagem original for FMXQH, os dados incluídos no relatório não incluem MQXQH. Os dados do relatório iniciam com o primeiro byte dos dados além do MQXQH na mensagem original. Isso ocorre se a fila for uma fila de transmissão..
5. Se uma mensagem de relatório de COA, COD ou expiração for recebida na fila de resposta, será garantido que a mensagem original chegou, foi entregue ou expirou, conforme apropriado. No entanto, se uma ou mais dessas mensagens de relatório forem solicitadas e não forem recebidas, o inverso não poderá ser assumido, pois pode ter ocorrido uma das seguintes situações:
  - a. A mensagem de relatório é retida porque um link está inativo
  - b. A mensagem de relatório é retida porque existe uma condição de bloqueio em uma fila de transmissão intermediária ou na fila de resposta (por exemplo, a fila está cheia ou inibida para puts)
  - c. A mensagem de relatório está em uma fila de mensagens não entregues
  - d. Quando o gerenciador de fila estava tentando gerar a mensagem de relatório, ele não pôde colocá-la na fila apropriada e também não conseguiu colocá-la na fila de mensagens não entregues, portanto, a mensagem de relatório não pôde ser gerada
  - e. Ocorreu uma falha do gerenciador de filas entre a ação sendo relatada (chegada, entrega ou expiração) e a geração da mensagem de relatório correspondente. (Isso não acontece para mensagens de relatório COD se o aplicativo recuperar a mensagem original em uma unidade de trabalho, pois a mensagem de relatório COD é gerada na mesma unidade de trabalho.)

As mensagens de relatório de exceções podem ser retidas da mesma maneira por motivos 1, 2 e 3 anteriormente. No entanto, quando um MCA não consegue gerar uma mensagem de relatório de exceção (a mensagem de relatório não pode ser colocada na fila de respostas ou na fila de mensagens não entregues), a mensagem original permanece na fila de transmissão no emissor e o canal é fechado. Isso ocorre independentemente de a mensagem de relatório ser gerada na extremidade de envio ou de recebimento do canal.

6. Se a mensagem original estiver temporariamente bloqueada (resultando na geração de uma mensagem de relatório de exceção e na colocação da mensagem original em uma fila de mensagens não entregues), mas o bloqueio for limpo e um aplicativo, em seguida, ler a mensagem original da fila de mensagens não entregues e colocá-la novamente em seu destino, poderá ocorrer o seguinte:
  - Mesmo que uma mensagem de relatório de exceção tenha sido gerada, a mensagem original eventualmente chega com êxito a seu destino
  - Mais de uma mensagem de relatório de exceção é gerada em relação a uma única mensagem original, já que a mensagem original pode encontrar outro bloqueio posteriormente.

#### **Mensagens de relatório ao colocar em um tópico:**

1. Os relatórios podem ser gerados ao colocar uma mensagem em um tópico. Esta mensagem será enviada para todos os assinantes do tópico, que pode ser zero, um ou muitos. Isso deve ser levado em consideração ao optar por usar opções de relatório, pois muitas mensagens de relatório poderiam ser geradas como resultado...
2. Ao colocar uma mensagem em um tópico, pode haver muitas filas de destino que devem receber uma cópia da mensagem. Se algumas dessas filas de destino tiverem um problema, como fila cheia, a conclusão bem-sucedida do MQPUT dependerá da configuração de NPMGDLV ou PMSGDLV (dependendo da persistência da mensagem). Se a configuração for tal que a entrega de mensagens para a fila de destino deve ser bem-sucedida (por exemplo, é uma mensagem

persistente para um assinante durável e PMSGDLV é configurado como ALL ou ALLDUR), o sucesso será definido como um dos seguintes critérios que estão sendo atendidos:

- Colocação bem-sucedida na fila de assinantes
- Uso de RODLQ e uma colocação bem-sucedida na fila de Devoluções se a fila de assinantes não puder receber a mensagem
- O uso de RODISC se a fila de assinantes não puder receber a mensagem

#### **Mensagens de relatório para segmentos de mensagens:**

1. Mensagens de relatório podem ser solicitadas para mensagens que tenham segmentação permitida (consulte a descrição do sinalizador MFSEGA). Se o gerenciador de filas achar necessário segmentar a mensagem, uma mensagem de relatório poderá ser gerada para cada um dos segmentos que subsequentemente encontrar a condição relevante. Os aplicativos devem, portanto, estar preparados para receber várias mensagens de relatório para cada tipo de mensagem de relatório solicitada. O campo MDGID na mensagem de relatório pode ser usado para correlacionar os diversos relatórios ao identificador de grupo da mensagem original e o campo MDFB usado para identificar o tipo de cada mensagem de relatório.
2. Se GMLOGO for usado para recuperar mensagens de relatório para segmentos, esteja ciente de que relatórios de tipos diferentes podem ser retornados pelas chamadas MQGET sucessivas. Por exemplo, se ambos os relatórios COA e COD forem solicitados para uma mensagem que é segmentada pelo gerenciador de filas, as chamadas MQGET para as mensagens de relatório poderão retornar as mensagens de relatório COA e COD intercaladas de uma maneira imprevisível. Isto pode ser evitado usando a opção GMCMPM (opcionalmente com GMATM). GMCMPM faz com que o gerenciador de filas remonte mensagens de relatório que possuem o mesmo tipo de relatório. Por exemplo, a primeira chamada MQGET pode remontar todas as mensagens COA relacionadas à mensagem original e a segunda chamada MQGET pode remontar todas as mensagens COD. O que é remontado primeiro depende do tipo de mensagem de relatório que ocorre primeiro na fila.
3. Os próprios aplicativos que colocam segmentos podem especificar diferentes opções de relatório para cada segmento. No entanto, os seguintes pontos devem ser observados:
  - Se os segmentos forem recuperados usando a opção GMCMPM, apenas as opções de relatório no primeiro segmento serão respeitadas pelo gerenciador de filas.
  - Se os segmentos forem recuperados um por um e a maioria deles tiver uma das opções ROCOD\*, mas pelo menos um segmento não, não será possível usar a opção GMCMPM para recuperar as mensagens de relatório com uma única chamada MQGET ou usar a opção GMASGA para detectar quando todas as mensagens de relatório tiverem chegado.
4. Em uma rede do MQ, é possível que os gerenciadores de filas tenham recursos diferentes. Se uma mensagem de relatório para um segmento for gerada por um gerenciador de fila ou MCA que não suporta segmentação, o gerenciador de filas ou MCA não incluirá, por padrão, as informações de segmento necessárias na mensagem de relatório e isso pode dificultar a identificação da mensagem original que causou a geração do relatório. Essa dificuldade pode ser evitada solicitando dados com a mensagem de relatório, ou seja especificando as opções RO\* D ou RO\* F apropriadas.. No entanto, esteja ciente de que, se RO\* D for especificado, menos de 100 bytes de dados da mensagem do aplicativo poderão ser retornados para o aplicativo que recupera a mensagem de relatório, se a mensagem de relatório for gerada por um gerenciador de filas ou MCA que não suporta a segmentação..

**Conteúdo do descritor de mensagens para uma mensagem de relatório:** Quando o gerenciador de fila ou o agente do canal de mensagens (MCA) gera uma mensagem de relatório, ele configura os campos no descritor de mensagem para os valores a seguir e, em seguida, coloca a mensagem de maneira normal..

*Tabela 708. Valores utilizados para campos MQMD quando uma mensagem de relatório é gerada pelo sistema*

<b>Campo no MQMD</b>	<b>Valor Usado</b>
MDSID	MDSIDV

Tabela 708. Valores utilizados para campos MQMD quando uma mensagem de relatório é gerada pelo sistema (continuação)

<b>Campo no MQMD</b>	<b>Valor Usado</b>
MDVER	MDVER2
MDREP	RONONE
MDMT	MTRPRT
MDEXP	EIULIM
MDFB	Conforme apropriado para a natureza do relatório (FBCOA, FBCOD, FBEXP ou um valor de RC*)
MDENC	Copiado do descritor de mensagens original
MDCSI	Copiado do descritor de mensagens original
MDFMT	Copiado do descritor de mensagens original
MDPRI	Copiado do descritor de mensagens original
MDPER	Copiado do descritor de mensagens original
MDMID	Conforme especificado pelas opções de relatório no descritor de mensagem original..
MDCID	Conforme especificado pelas opções de relatório no descritor de mensagem original..
MDBOC	0
MDRQ	Espaços em branco
MDRM	Nome do gerenciador de filas
MDUID	Conforme definido pela opção PMPASI
MDACC	Conforme definido pela opção PMPASI
MDAID	Conforme definido pela opção PMPASI
MDPAT	ATQM ou conforme apropriado para o agente do canal de mensagens
MDPAN	Primeiros 28 bytes do nome do gerenciador de filas ou nome do agente do canal de mensagens. Para mensagens de relatório geradas pela ponte IMS , esse campo contém o nome do grupo XCF e o nome do membro XCF do sistema IMS ao qual a mensagem se relaciona
MDPD	Data quando a mensagem de relatório é enviada
MDPT	Hora em que a mensagem de relatório é enviada..
MDAOD	Espaços em branco
MDGID	Copiado do descritor de mensagens original
MDSEQ	Copiado do descritor de mensagens original
MDOFF	Copiado do descritor de mensagens original
MDMFL	Copiado do descritor de mensagens original
MDOLN	Copiado do descritor de mensagens original, se não OLUNDF, e configurado para o comprimento dos dados da mensagem original, caso contrário

Um aplicativo que gera um relatório é recomendado para configurar valores semelhantes, exceto para o seguinte:



- O campo MDRM pode ser definido como espaços em branco (o gerenciador de filas alterará isso para o nome do gerenciador de fila local quando a mensagem for colocada)..
- Os campos de contexto devem ser configurados usando a opção que seria usada para uma resposta, normalmente PMPASI.

**Analisando o campo de relatório:** o campo MDREP contém subcampos; por isso, os aplicativos que precisam verificar se o emissor da mensagem solicitou um relatório específico devem usar uma das técnicas descritas em [“Analisando o campo de relatório em IBM i ..”](#) na página 1486.

Esse é um campo de saída para a chamada MQGET e um campo de entrada para as chamadas MQPUT e MQPUT1 . O valor inicial deste campo é RONONE..

### **MDRM (sequência de caracteres de 48 bytes)**

Nome do gerenciador de filas de resposta.

Este é o nome do gerenciador de filas para o qual a mensagem de resposta ou de relatório deve ser enviada. MDRQ é o nome local de uma fila definida neste gerenciador de filas.

Se o campo MDRM estiver em branco, o gerenciador de filas locais consulta o nome **MDRQ** em suas definições de fila. Se existir uma definição local de uma fila remota com esse nome, o valor **MDRM** na mensagem transmitida será substituído pelo valor do atributo **RemoteQMgrName** da definição da fila remota e esse valor será retornado no descritor de mensagens quando o aplicativo de recebimento emitir uma chamada MQGET para a mensagem. Se uma definição local de uma fila remota não existir, o MDRM transmitido com a mensagem será o nome do gerenciador de filas locais.

Se o nome for especificado, ele poderá conter espaços em branco finais; o primeiro caractere nulo e os caracteres seguintes serão tratados como espaços em branco. Caso contrário, no entanto, não será feita nenhuma verificação de que o nome satisfaz as regras de nomenclatura para gerenciadores de filas ou que esse nome é conhecido para o gerenciador de filas de envio; isso também é verdadeiro para o nome transmitido, se o **MDRM** for substituído na mensagem transmitida.

Se uma fila de resposta não for necessária, é recomendado (embora isso não esteja marcado) que o campo MDRM seja configurado como espaços em branco; o campo não deve ser deixado sem inicialização.

Para a chamada MQGET, o gerenciador de filas sempre retorna o nome preenchido com espaços em branco no comprimento do campo.

Esse é um campo de saída para a chamada MQGET e um campo de entrada para as chamadas MQPUT e MQPUT1 . O comprimento deste campo é fornecido por LNQMN O valor inicial desse campo é 48 caracteres em branco.

### **MDRQ (sequência de caracteres de 48 bytes)**

Nome da fila de resposta.

Este é o nome da fila de mensagens para a qual o aplicativo que emitiu o pedido de obtenção para a mensagem deve enviar mensagens MTRPLY e MTRPRT.. O nome é o nome local de uma fila definida no gerenciador de filas identificado por MDRM. Essa fila não deve ser uma fila de modelo, embora o gerenciador de filas de envio não verifique isso quando a mensagem é colocada

Para as chamadas MQPUT e MQPUT1 , esse campo não deve ficar em branco se o campo MDMT tiver o valor MTRQST ou se alguma mensagem de relatório for solicitada pelo campo MDREP . No entanto, o valor especificado (ou substituído) é transmitido para o aplicativo que emite a solicitação get para a mensagem, independentemente do tipo de mensagem.

Se o campo MDRM estiver em branco, o gerenciador de filas locais consulta o nome MDRQ em suas definições de fila. Se existir uma definição local de uma fila remota com esse nome, o valor MDRQ na mensagem transmitida será substituído pelo valor do atributo **RemoteQName** da definição da fila remota e esse valor será retornado no descritor de mensagens quando o aplicativo de recebimento emitir uma chamada MQGET para a mensagem. Se uma definição local de uma fila remota não existir, MDRQ permanecerá inalterado

Se o nome for especificado, ele poderá conter espaços em branco finais; o primeiro caractere nulo e os caracteres seguintes serão tratados como espaços em branco. Caso contrário, no entanto, não será feita nenhuma verificação de que o nome satisfaz as regras de nomenclatura para filas; isso também será verdadeiro para o nome transmitido, se o MDRQ for substituído na mensagem transmitida. A única verificação feita é que um nome foi especificado, se as circunstâncias o exigirem.

Se uma fila de resposta não for necessária, é recomendado (embora isso não esteja marcado) que o campo MDRQ seja configurado como espaços em branco; o campo não deve ser deixado sem inicialização.

Para a chamada MQGET, o gerenciador de filas sempre retorna o nome preenchido com espaços em branco no comprimento do campo.

Se uma mensagem que requer uma mensagem de relatório não puder ser entregue e a mensagem de relatório também não puder ser entregue na fila especificada, a mensagem original e a mensagem de relatório vão para a fila de mensagens não entregues (mensagens não entregues). Consulte o atributo **DeadLetterQName** descrito em [“Atributos para o gerenciador de filas no IBM i”](#) na página 1449..

Esse é um campo de saída para a chamada MQGET e um campo de entrada para as chamadas MQPUT e MQPUT1 . O comprimento deste campo é fornecido por LNQN (LNQN). O valor inicial desse campo é 48 caracteres em branco.

### **MDSEQ (número inteiro assinado de 10 dígitos)**

Número de sequência da mensagem lógica dentro do grupo.

Os números de sequência começam em 1 e aumentam em 1 para cada nova mensagem lógica no grupo, até um máximo de 999 999 999. Uma mensagem física que não está em um grupo tem um número de sequência de 1

Este campo não precisa ser configurado pelo aplicativo na chamada MQPUT ou MQGET se:

- Na chamada MQPUT, PMLOGO é especificado..
- Na chamada MQGET, MOSEQN não foi especificado..

Estas são as maneiras recomendadas de usar estas chamadas para mensagens que não são mensagens de relatório. No entanto, se o aplicativo requerer mais controle ou a chamada for MQPUT1, o aplicativo deverá assegurar que MDSEQ seja configurado para um valor apropriado.

Na entrada para as chamadas MQPUT e MQPUT1 , o gerenciador de filas usa o valor detalhado na [Tabela 1](#). Na saída das chamadas MQPUT e MQPUT1 , o gerenciador de filas configura esse campo para o valor que foi enviado com a mensagem..

Na entrada para a chamada MQGET, o gerenciador de filas usa o valor detalhado na [Tabela 1](#). Na saída da chamada MQGET, o gerenciador de filas configura esse campo para o valor da mensagem recuperada.

O valor inicial deste campo é um. Esse campo será ignorado se MDVER for menor que MDVER2.

### **MDSID (cadeia de caracteres de 4 bytes).**

Identificador de estruturação.

O valor deve ser:.

#### **MDSIDV**

Identificador para a estrutura do descritor de mensagem

Isto é sempre um campo de entrada. O valor inicial desse campo é MDSIDV.

### **MDUID (cadeia de caracteres de 12 bytes)**

Identificador do usuário.






Isso faz parte do *contexto de identidade* da mensagem Para obter mais informações sobre o contexto de mensagem, consulte [Contexto da mensagem](#) e [Informações de contexto de controle](#).

MDUID especifica o identificador de usuário do aplicativo que originou a mensagem. O gerenciador de filas trata essas informações como dados de caractere, mas não define o formato deles.

Após uma mensagem ter sido recebida, o MDUID poderá ser usado no campo ODAU do parâmetro **OBJDSC** de uma chamada MQOPEN ou MQPUT1 subsequente, para que a verificação de autorização seja executada para o usuário MDUID em vez de o aplicativo executar a abertura..

Quando o gerenciador de fila gera essas informações para uma chamada MQPUT ou MQPUT1, o gerenciador de filas usa um identificador de usuário determinado do ambiente.

Quando o identificador de usuários é determinado a partir do ambiente:

-  No z/OS, o gerenciador de filas usa:
  - Para lote, o identificador de usuário do cartão JES JOB ou da tarefa iniciada
  - Para TSO, o identificador de usuário de logon
  - Para CICS, o identificador de usuários associado à tarefa
  - Para IMS, o identificador de usuários depende do tipo de aplicativo:
    - para:
      - Regiões BMP sem mensagem
      - Regiões IFP sem mensagem
      - BMP de mensagem e regiões IFP de mensagem que não emitiram uma chamada GU bem-sucedida
    - o gerenciador de filas usa o identificador de usuário do cartão JES JOB da região ou o identificador de usuário TSO. Se estiverem em branco ou nulo, ele usará o nome do bloco de especificação de programa (PSB).
    - para:
      - BMP de mensagem e regiões IFP de mensagem que emitiram uma chamada de GU bem-sucedida
      - Regiões MPP
  - o gerenciador de filas usa um dos seguintes:
    - O identificador de usuário conectado associado à mensagem
    - O nome do terminal lógico (LTERM)
    - O identificador de usuários do cartão JES JOB da região
    - O identificador de usuário do TSO
    - O nome do PSB
-  No IBM i, o gerenciador de filas usa o nome do perfil do usuário associado à tarefa do aplicativo
-   No AIX and Linux, o gerenciador de filas usa:
  - O nome de logon do aplicativo
  - O identificador de usuário efetivo do processo se nenhum logon estiver disponível
  - O identificador de usuário associado à transação, se o aplicativo for uma transação CICS
- Em VSE/ESA, este é um campo reservado
-  No Windows, o gerenciador de filas usa os primeiros 12 caracteres do nome do usuário conectado.

Para as chamadas MQPUT e MQPUT1, este é um campo de entrada / saída se PMSETI ou PMSETA for especificado no parâmetro **PMO**. Quaisquer informações após um caractere nulo no campo são descartadas. O caractere nulo e quaisquer caracteres a seguir são convertidos em branco pelo

gerenciador de filas. Se PMSETI ou PMSETA não for especificado, este campo será ignorado na entrada e será um campo de saída.

Após a conclusão bem-sucedida de uma chamada MQPUT ou MQPUT1, esse campo contém o MDUID que foi transmitido com a mensagem se ele foi colocado em uma fila. Esse será o valor de MDUID que será mantido com a mensagem se ela for retida (consulte a descrição de PMRET para obter mais detalhes sobre as publicações retidas), mas não será usado como o MDUID quando a mensagem for enviada como uma publicação para os assinantes, pois eles fornecem um valor para substituir MDUID em todas as publicações enviadas para eles. Se a mensagem não tiver nenhum contexto, o campo estará inteiramente em branco.

Este é um campo de saída para a chamada MQGET. O comprimento deste campo é fornecido pelo LNUID. O valor inicial desse campo é de 12 caracteres em branco.

### MDVER (número inteiro assinado de 10 dígitos)

Número de versão da estrutura.

O valor deve ser um dos seguintes:

#### MDVER1

Estrutura do descritor de mensagens Version-1

#### MDVER2

Estrutura do descritor de mensagens Version-2.

**Nota:** Quando um MQMD version-2 é usado, o gerenciador de filas executa verificações adicionais em qualquer estrutura de cabeçalho do MQ que possa estar presente no início dos dados da mensagem do aplicativo; para obter detalhes adicionais, consulte as notas de uso para a chamada MQPUT.

Os campos que existem apenas na versão mais recente da estrutura são identificados como tal nas descrições dos campos. A constante a seguir especifica o número da versão atual:

#### MDVERC

A versão atual da estrutura do descritor de mensagem

Isto é sempre um campo de entrada. O valor inicial desse campo é MDVER1.

## Valores iniciais

<i>Tabela 709. Campos em MQMD</i>		
Nome de Campo	Nome da constante	Valor da constante
MDSID	MDSIDV	'MD--'
MDVER	MDVER1	1
MDREP	RONONE	0
MDMT	MTDGRM	8
MDEXP	EIULIM	-1
MDFB	FBNONE	0
MDENC	ENNAT	Depende do ambiente
MDCSI	CSQM	0
MDFMT	FMNONE	Espaços em branco
MDPRI	PRQDEF	-1
MDPER	PEQDEF	2
MDMID	MINONE	Nulos

Tabela 709. Campos em MQMD (continuação)

Nome de Campo	Nome da constante	Valor da constante
MDCID	CINONE	Nulos
MDBOC	Nenhum	0
MDRQ	Nenhum	Espaços em branco
MDRM	Nenhum	Espaços em branco
MDUID	Nenhum	Espaços em branco
MDACC	ACNONE	Nulos
MDAID	Nenhum	Espaços em branco
MDPAT	ATNCON	0
MDPAN	Nenhum	Espaços em branco
MDPD	Nenhum	Espaços em branco
MDPT	Nenhum	Espaços em branco
MDAOD	Nenhum	Espaços em branco
MDGID	GINONE	Nulos
MDSEQ	Nenhum	1
MDOFF	Nenhum	0
MDMFL	MFNONE	0
MDOLN	OLUNDF	-1
<b>Notas:</b>		
1. O símbolo - representa um único caractere em branco.		

## Declaração RPG

```

D*..1.....2.....3.....4.....5.....6.....7..
D*
D* MQMD Structure
D*
D* Structure identifier
D MDSID          1      4  INZ('MD ')
D* Structure version number
D MDVER          5      8I 0 INZ(1)
D* Options for report messages
D MDREP          9     12I 0 INZ(0)
D* Message type
D MDMT          13     16I 0 INZ(8)
D* Message lifetime
D MDEXP         17     20I 0 INZ(-1)
D* Feedback or reason code
D MDFB          21     24I 0 INZ(0)
D* Numeric encoding of message data
D MDENC         25     28I 0 INZ(273)
D* Character set identifier of messagedata
D MDCSI         29     32I 0 INZ(0)
D* Format name of message data
D MDFMT         33     40  INZ(' ')
D* Message priority
D MDPRI         41     44I 0 INZ(-1)
D* Message persistence
D MDPER         45     48I 0 INZ(2)
D* Message identifier
D MDMID         49     72  INZ(X'0000000000000000-

```

```

D          000000000000000000000000-
D          000000000000')
D* Correlation identifier
D MDCID          73      96      INZ(X'00000000000000-
D          00000000000000000000-
D          000000000000')
D* Backout counter
D MDBC          97      100I 0 INZ(0)
D* Name of reply queue
D MDRQ          101     148      INZ
D* Name of reply queue manager
D MDRM          149     196      INZ
D* User identifier
D MDUID         197     208      INZ
D* Accounting token
D MDACC         209     240      INZ(X'00000000000000-
D          00000000000000000000-
D          000000000000000000-
D          000000')
D* Application data relating to identity
D MDAID         241     272      INZ
D* Type of application that put the message
D MDPAT         273     276I 0 INZ(0)
D* Name of application that put the message
D MDPAN         277     304      INZ
D* Date when message was put
D MDPD          305     312      INZ
D* Time when message was put
D MDPT          313     320      INZ
D* Application data relating to origin
D MDAOD         321     324      INZ
D* Group identifier
D MDGID         325     348      INZ(X'00000000000000-
D          00000000000000000000-
D          000000000000')
D* Sequence number of logical message within group
D MDSEQ         349     352I 0 INZ(1)
D* Offset of data in physical message from start of logical message
D MDOFF         353     356I 0 INZ(0)
D* Message flags
D MDMFL         357     360I 0 INZ(0)
D* Length of original message
D MDOLN         361     364I 0 INZ(-1)

```

## IBM i MQMDE (extensão do descritor de mensagens) em IBM i

### Visão Geral

**Propósito:** A estrutura MQMDE descreve os dados que às vezes ocorrem antes dos dados da mensagem do aplicativo. A estrutura contém os campos MQMD que existem no MQMD version-2, mas não no MQMD version-1.

**Nome do Formato:** FMMDE

**Conjunto de caracteres e codificação:** os dados em MQMDE devem estar no conjunto de caracteres fornecido pelo atributo do gerenciador de filas **CodedCharSetId** e a codificação do gerenciador de filas locais fornecido por ENNAT para a linguagem de programação C.

O conjunto de caracteres e a codificação do MQMDE devem ser configurados nos campos *MDCSI* e *MDENC* em:

- O MQMD (se a estrutura MQMDE estiver no início dos dados da mensagem) ou
- A estrutura do cabeçalho que precede a estrutura MQMDE (todos os outros casos).

Se o MQMDE não estiver no conjunto de caracteres e na codificação do gerenciador de filas, o MQMDE será aceito, mas não honrado, ou seja, o MQMDE será tratado como dados da mensagem

**Uso:** os aplicativos normais devem usar um MQMD version-2, caso em que não encontrarão uma estrutura MQMDE. No entanto, aplicativos especializados e aplicativos que continuam usando um MQMD version-1 podem encontrar um MQMDE em algumas situações. A estrutura MQMDE pode ocorrer nas seguintes circunstâncias:

- Especificado nas chamadas MQPUT e MQPUT1
- Retornado pela chamada MQGET
- Em mensagens em filas de transmissão
- “MQMDE especificado em chamadas MQPUT e MQPUT1 .” na página 1195
- “MQMDE retornado pela chamada MQGET” na página 1195
- “MQMDE em mensagens em filas de transmissão” na página 1196
- “Campos” na página 1196
- “Valores iniciais” na página 1198
- “Declaração RPG” na página 1199

## **MQMDE especificado em chamadas MQPUT e MQPUT1 .**

Nas chamadas MQPUT e MQPUT1 , se o aplicativo fornecer um MQMD version-1 , o aplicativo poderá opcionalmente prefixar os dados da mensagem com um MQMDE, configurando o campo *MDFMT* em MQMD para FMMDE para indicar que um MQMDE está presente. Se o aplicativo não fornecer um MQMDE, o gerenciador de filas assume valores padrão para os campos no MQMDE. Os valores padrão que o gerenciador de filas usa são iguais aos valores iniciais para a estrutura-consulte [Tabela 711 na página 1198](#).

Se o aplicativo fornecer um version-2 MQMD e prefixar os dados da mensagem do aplicativo com um MQMDE, as estruturas serão processadas conforme mostrado em [Tabela 710 na página 1195](#)

<i>Tabela 710. Ação do gerenciador de filas quando MQMDE especificado em MQPUT ou MQPUT1</i>			
<b>MQMD Versão</b>	<b>Valores de campos version-2</b>	<b>Valores de campos correspondentes em MQMDE</b>	<b>Ação executada pelo gerenciador da fila</b>
1	-	Válidos	MQMDE é honrado
2	Padrão	Válidos	MQMDE é honrado
2	Não padrão	Válidos	MQMDE é tratado como dados da mensagem
1 ou 2	Qualquer	Inválido	A chamada falha com um código de razão apropriado..
1 ou 2	Qualquer	MQMDE está no conjunto de caracteres ou na codificação errada ou é uma versão não suportada	MQMDE é tratado como dados da mensagem

Há um caso especial. Se o aplicativo usar um MQMD version-2 para colocar uma mensagem que é um segmento (ou seja, o sinalizador MFSEG ou MFLSEG está configurado) e o nome do formato no MQMD for FMDLH, o gerenciador de filas gerará uma estrutura MQMDE e o inserirá *entre* a estrutura MQDLH e os dados que a seguem No MQMD que o gerenciador de filas retém com a mensagem, os campos version-2 são configurados para seus valores padrão

Vários dos campos existentes no MQMD version-2 , mas não no MQMD version-1 são campos de entrada / saída em MQPUT e MQPUT1. No entanto, o gerenciador de filas não retorna nenhum valor nos campos equivalentes no MQMDE na saída das chamadas MQPUT e MQPUT1 ; se o aplicativo requerer esses valores de saída, ele deverá usar um MQMD version-2 .

## **MQMDE retornado pela chamada MQGET**

Na chamada MQGET, se o aplicativo fornecer um MQMD version-1 , o gerenciador de filas prefixa a mensagem retornada com um MQMDE, mas somente se um ou mais dos campos no MQMDE tiver um

valor não padrão. O gerenciador de filas configura o campo *MDFMT* no MQMD para o valor FMMDE para indicar que um MQMDE está presente..

Se o aplicativo fornecer um MQMDE no início do parâmetro **BUFFER** , o MQMDE será ignorado No retorno da chamada MQGET, ele é substituído pelo MQMDE para a mensagem (se for necessário) ou sobrescrito pelos dados da mensagem do aplicativo (se o MQMDE não for necessário).

Se um MQMDE for retornado pela chamada MQGET, os dados no MQMDE geralmente estão no conjunto e na codificação de caracteres do gerenciador de filas No entanto, o MQMDE pode estar em algum outro conjunto de caracteres e codificação se:

- O MQMDE foi tratado como dados na chamada MQPUT ou MQPUT1 (consulte [Tabela 710 na página 1195](#) para obter as circunstâncias que podem causar isso).
- A mensagem foi recebida de um gerenciador de filas remotas conectado por uma conexão TCP e o agente do canal de mensagens de recebimento (MCA) não foi configurado corretamente (consulte [Segurança de IBM MQ for IBM i objetos](#) para obter informações adicionais).

## MQMDE em mensagens em filas de transmissão

Mensagens em filas de transmissão são prefixadas com a estrutura MQXQH, que contém dentro dela um MQMD version-1 . Um MQMDE também pode estar presente, posicionado entre a estrutura MQXQH e os dados da mensagem do aplicativo, mas geralmente estará presente apenas se um ou mais dos campos no MQMDE tiverem um valor não padrão.

Outras estruturas de cabeçalho do IBM MQ também podem ocorrer entre a estrutura MQXQH e os dados da mensagem do aplicativo Por exemplo, quando o cabeçalho MQDLH de devoluções está presente e a mensagem não é um segmento, a ordem é:

- MQXQH (contendo um MQMD version-1 )
- MQMDE
- MQDLH
- Dados da mensagem do aplicativo

## Campos

A estrutura MQMDE contém os seguintes campos; os campos são descritos em **ordem alfabética**:

### MECSI (número inteiro assinado de 10 dígitos)

Identificador de conjunto de caracteres de dados que segue MQMDE.

Isso especifica o identificador do conjunto de caracteres dos dados que seguem a estrutura MQMDE; ele não se aplica aos dados de caracteres na própria estrutura MQMDE

Na chamada MQPUT ou MQPUT1 , o aplicativo deve configurar esse campo para o valor apropriado aos dados. O gerenciador de filas não verifica se esse campo é válido O seguinte valor especial pode ser usado:

#### CSINHT

Herde o identificador do conjunto de caracteres desta estrutura..

Os dados de caractere nos dados *após* essa estrutura estão no mesmo conjunto de caracteres que essa estrutura

O gerenciador de filas altera esse valor na estrutura enviada na mensagem para o identificador do conjunto de caractere real da estrutura.. Se nenhum erro ocorrer, o valor CSINHT não será retornado pela chamada MQGET.

CSINHT não poderá ser usado se o valor do campo *MDPAT* em MQMD for ATBRKR.

O valor inicial desse campo é CSUNDF.

### MEENC (número inteiro assinado de 10 dígitos)

MEENC (número inteiro assinado de 10 dígitos)



Isso especifica a codificação numérica dos dados que seguem a estrutura MQMDE; ela não se aplica a dados numéricos na própria estrutura MQMDE

Na chamada MQPUT ou MQPUT1 , o aplicativo deve configurar esse campo para o valor apropriado aos dados. O gerenciador de filas não verifica se o campo é válido.. Consulte o campo *MDENC* descrito em [“MQMD \(Descritor de mensagens\) em IBM i” na página 1148](#) para obter mais informações sobre as codificações de dados

O valor inicial deste campo é ENNAT

#### **MEFLG (número inteiro assinado de 10 dígitos)**

Sinalizadores gerais

A seguinte sinalização pode ser especificada:

##### **MEFNON**

Sem bandeiras.

O valor inicial deste campo é MEFNON.

#### **MEFMT (cadeia de caracteres de 8 bytes).**

Nome do formato de dados que seguem o MQMDE

Especifica o nome do formato dos dados que seguem a estrutura MQMDE.

Na chamada MQPUT ou MQPUT1 , o aplicativo deve configurar esse campo para o valor apropriado aos dados. O gerenciador de filas não verifica se esse campo é válido Consulte o campo *MDFMT* descrito em [“MQMD \(Descritor de mensagens\) em IBM i” na página 1148](#) para obter mais informações sobre os nomes de formato

O valor inicial deste campo é FMNONE.

#### **MEGID (cadeia de bits de 24 bytes)**

Identificador de grupo.

Consulte o campo *MDGID* descrito em [“MQMD \(Descritor de mensagens\) em IBM i” na página 1148](#).. O valor inicial deste campo é GINONE.

#### **MELEN (número inteiro assinado de 10 dígitos)**

Comprimento da estrutura do MQMDE

O seguinte valor é definido:

##### **MELEN2**

Comprimento da estrutura de extensão do descritor de mensagens version-2

O valor inicial deste campo é MELEN2.

#### **MEMFL (número inteiro assinado de 10 dígitos)**

Sinalizadores de mensagem

Consulte o campo *MDMFL* descrito em [“MQMD \(Descritor de mensagens\) em IBM i” na página 1148](#).. O valor inicial deste campo é MFNONE.

#### **MEOFF (número inteiro assinado de 10 dígitos)**

Deslocamento dos dados na mensagem física a partir do início da mensagem lógica.

Consulte o campo *MDOFF* descrito em [“MQMD \(Descritor de mensagens\) em IBM i” na página 1148](#).. O valor inicial desse campo é 0.

#### **MEOLN (número inteiro assinado de 10 dígitos)**

Comprimento da mensagem original.

Consulte o campo *MDOLN* descrito em [“MQMD \(Descritor de mensagens\) em IBM i” na página 1148](#).. O valor inicial deste campo é OLUNDF.

### MESEQ (número inteiro assinado de 10 dígitos)

Número de sequência da mensagem lógica dentro do grupo.

Consulte o campo *MDSEQ* descrito em “MQMD (Descritor de mensagens) em IBM i” na página 1148.. O valor inicial desse campo é 1.

### MESID (sequência de caracteres de 4 bytes).

Identificador de estruturação.

O valor deve ser:.

#### MESIDV

Identificador da estrutura de extensão do descritor de mensagens.

O valor inicial desse campo é MESIDV.

### MEVER (número inteiro assinado de 10 dígitos)

Número de versão da estrutura.

O valor deve ser:.

#### MEVER2

Estrutura da extensão do descritor de mensagens Version-2

A constante a seguir especifica o número da versão atual:

#### MEVERC

A versão atual da estrutura de extensão do descritor de mensagem

O valor inicial desse campo é MEVER2.

## Valores iniciais

Tabela 711. Campos em MQMDE		
Nome de Campo	Nome da constante	Valor da constante
<i>MESID</i>	MESIDV	'MDE↵'
<i>MEVER</i>	MEVER2	2
<i>MELEN</i>	MELEN2	72
<i>MEENC</i>	ENNAT	Depende do ambiente
<i>MECSI</i>	CSUNDF	0
<i>MEFMT</i>	FMNONE	Espaços em branco
<i>MEFLG</i>	MEFNON	0
<i>MEGID</i>	GINONE	Nulos
<i>MESEQ</i>	Nenhum	1
<i>MEOFF</i>	Nenhum	0
<i>MEMFL</i>	MFNONE	0
<i>MEOLN</i>	OLUNDF	-1

**Notas:**

1. O símbolo ↵ representa um único caractere em branco.

## Declaração RPG

```
D*..1....:....2....:....3....:....4....:....5....:....6....:....7..
D*
D* MQMDE Structure
D*
D* Structure identifier
D MESID          1          4  INZ('MDE ')
D* Structure version number
D MEVER          5          8I 0 INZ(2)
D* Length of MQMDE structure
D MELEN          9         12I 0 INZ(72)
D* Numeric encoding of data that followsMQMDE
D MEENC         13         16I 0 INZ(273)
D* Character-set identifier of data thatfollows MQMDE
D MECSI         17         20I 0 INZ(0)
D* Format name of data that followsMQMDE
D MEFMT         21         28  INZ('      ')
D* General flags
D MEFLG         29         32I 0 INZ(0)
D* Group identifier
D MEGID         33         56  INZ('X'0000000000000000-
D                                     000000000000000000000000-
D                                     000000000000')
D* Sequence number of logical messagewithin group
D MESEQ         57         60I 0 INZ(1)
D* Offset of data in physical messagefrom start of logical message
D MEOFF         61         64I 0 INZ(0)
D* Message flags
D MEMFL         65         68I 0 INZ(0)
D* Length of original message
D MEOLN         69         72I 0 INZ(-1)
```

## IBM i MQMHBO (Identificador de mensagens para opções de buffer) em IBM i

Estrutura que define o identificador de mensagens para opções de buffer...

### Visão Geral

**Propósito:** A estrutura MQMHBO permite que os aplicativos especifiquem opções que controlam como buffers são produzidos a partir de identificadores de mensagens. A estrutura é um parâmetro de entrada na chamada MQMHBUF..

**Conjunto de caracteres e codificação:** os dados em MQMHBO devem estar no conjunto de caracteres do aplicativo e codificação do aplicativo (ENNAT).

- [“Campos” na página 1199](#)
- [“Valores iniciais” na página 1200](#)
- [“Declaração RPG” na página 1200](#)

### Campos

A estrutura MQMHBO contém os campos a seguir; os campos são descritos em **ordem alfabética**:

#### MBOPT (número inteiro assinado de 10 dígitos)

Identificador de mensagem para estrutura de opções de buffer-campo MBOPT.

Essas opções controlam a ação de MQMHBUF.

Você deve especificar a seguinte opção:

#### MBPRRF

Ao converter propriedades de uma manipulação de mensagem em um buffer, converta-as no formato MQRFH2 .

Opcionalmente, também é possível especificar a opção a seguir: Para especificar mais de uma opção, inclua os valores juntos (não inclua a mesma constante mais de uma vez) ou combine os valores usando a operação OR bit a bit (se a linguagem de programação suportar operações de bit).

#### **MBDLPR**

As propriedades incluídas no buffer são excluídas do identificador de mensagem. Se a chamada falhar nenhuma propriedade será excluída.

Isto é sempre um campo de entrada. O valor inicial deste campo é MBPRRF.

#### **MBSID (número inteiro assinado de 10 dígitos)**

Identificador de mensagem para a estrutura de opções de buffer-campo MBSID.

Este é o identificador de estrutura. O valor deve ser:.

#### **MBSIDV**

Identificador da manipulação de mensagens para a estrutura de opções de buffer.

Isto é sempre um campo de entrada. O valor inicial desse campo isMBSIDV..

#### **MBVER (número inteiro assinado de 10 dígitos)**

Este é o número da versão da estrutura O valor deve ser:.

#### **MBVER1**

Número da versão para a manipulação de mensagens para a estrutura de opções de buffer

A constante a seguir especifica o número da versão atual:

#### **MBVERC**

Versão atual da manipulação de mensagem para a estrutura de opções de buffer

Isto é sempre um campo de entrada. O valor inicial desse campo é MBVER1.

### **Valores iniciais**

<i>Tabela 712. Campos em MQMHBO</i>		
<b>Nome de Campo</b>	<b>Nome da constante</b>	<b>Valor da constante</b>
<i>MVSID</i>	MBSIDV	'MHBO'
<i>MBVER</i>	MBVER1	1
<i>MBOPT</i>	MBPRRF	

#### **Notas:**

1. O valor Sequência nula ou espaços em branco indica um caractere em branco.

### **Declaração RPG**

```

D* MQMHBO Structure
D*
D*
D* Structure identifier
D MBSID          1      4   INZ('MHBO')
D*
D* Structure version number
D MBVER          5      8I 0 INZ(1)
D*
D* Options that control the action of MQMHBUF
D MBOPT          9     12I 0 INZ(1)

```

### **MQOD (descriptor de objeto) em IBM i**

A estrutura MQOD é usada para especificar um objeto por nome..

## Visão Geral

**Propósito:** Os seguintes tipos de objeto são válidos:

- Fila ou lista de distribuição
- Lista de Nomes
- Definição de processo
- Gerenciador de filas
- Tópico

A estrutura é um parâmetro de entrada / saída nas chamadas MQOPEN e MQPUT1 .

**Versão:** A versão atual do MQOD é ODVER4 Os campos que existem apenas nas versões mais recentes da estrutura são identificados como tal nas descrições a seguir:

O arquivo COPY fornecido contém a versão mais recente do MQOD suportada pelo ambiente, mas com o valor inicial do campo *ODVER* configurado como ODVER1. Para usar campos que não estão presentes na estrutura version-1 , o aplicativo deve configurar o campo *ODVER* para o número da versão necessária.

Para abrir uma lista de distribuição, *ODVER* deve ser ODVER2 ou superior.

**Conjunto de caracteres e codificação:** os dados em MQOD devem estar no conjunto de caracteres fornecido pelo atributo do gerenciador de filas **CodedCharSetId** e a codificação do gerenciador de fila local fornecido por ENNAT. No entanto, se o aplicativo estiver em execução como um cliente IBM MQ , a estrutura deverá estar no conjunto de caracteres e na codificação do cliente..

- [“Campos” na página 1201](#)
- [“Valores iniciais” na página 1208](#)
- [“Declaração RPG” na página 1209](#)

## Campos

A estrutura MQOD contém os seguintes campos; os campos são descritos em **ordem alfabética**:

### **ODASI (sequência de bits de 40 bytes)**

Identificador de segurança alternativo.

Este é um identificador de segurança que é transmitido com o *ODAU* para o serviço de autorização para permitir que verificações de autorização apropriadas sejam executadas. *ODASI* é usado apenas se:

- OOALTU é especificado na chamada MQOPEN ou
- PMALTU é especificado na chamada MQPUT1 ,

e o campo *ODAU* não está totalmente em branco até o primeiro caractere nulo ou o final do campo..

O campo *ODASI* possui a seguinte estrutura:

- O primeiro byte é um número inteiro binário contendo o comprimento dos dados significativos que seguem; o valor exclui o próprio byte de comprimento. Se não houver nenhum identificador de segurança presente, o comprimento será zero.
- O segundo byte indica o tipo de identificador de segurança que está presente; os valores a seguir são possíveis:

#### **SITWNT**

Identificador de segurança Windows .

#### **SITNON**

Não há identificador de segurança.

- O terceiro e os bytes subsequentes até o comprimento definido pelo primeiro byte contêm o próprio identificador de segurança.
- Os bytes restantes no campo são configurados como binários zero.

O valor especial a seguir pode ser usado:

**SINONE**

Nenhum identificador de segurança especificado.

O valor é binário zero para o comprimento do campo.

Este é um campo de entrada. O comprimento desse campo é fornecido pelo LNSCID.. O valor inicial deste campo é SINONE. Este campo será ignorado se *ODVER* for menor que *ODVER3*..

**ODAU (sequência de caracteres de 12 bytes)**

Identificador de usuário alternativo

Se *OOALTU* for especificado para a chamada *MQOPEN* ou *PMALTU* para a chamada *MQPUT1*, esse campo contém um identificador de usuário alternativo que deve ser usado para verificar a autorização para a abertura, no lugar do identificador de usuário sob o qual o aplicativo está atualmente em execução. No entanto, algumas verificações ainda são realizadas com o identificador de usuário atual (por exemplo, verificações de contexto).

Se *OOALTU* e *PMALTU* não forem especificados e este campo estiver totalmente em branco até o primeiro caractere nulo ou o final do campo, a abertura só poderá ser bem-sucedida se nenhuma autorização do usuário for necessária para abrir este objeto com as opções especificadas..

Se nem *OOALTU* nem *PMALTU* forem especificados, este campo será ignorado

Este é um campo de entrada. O comprimento deste campo é fornecido pelo *LNUID*. O valor inicial desse campo é de 12 caracteres em branco.

**ODDN (sequência de caracteres de 48 bytes).**

Nome da fila dinâmica..

Esse é o nome de uma fila dinâmica que deve ser criada pela chamada de *MQOPEN*. Isso é relevante apenas quando *ODON* especifica o nome de uma fila modelo; em todos os casos, *ODDN* é ignorado.

Os caracteres que são válidos no nome são os mesmos para *ODON*, exceto que um asterisco também é válido. Um nome que está em branco (ou um no qual apenas espaços em branco são mostrados antes do primeiro caractere nulo) não será válido se *ODON* for o nome de uma fila modelo...

Se o último caractere não em branco no nome for um asterisco ( \* ), o gerenciador de filas substituirá o asterisco por uma cadeia de caracteres que garante que o nome gerado para a fila seja exclusivo no gerenciador de fila local. Para permitir um número suficiente de caracteres para isso, o asterisco é válido apenas nas posições de 1 a 33. Não deve haver caracteres diferentes de espaços em branco ou um caractere nulo após o asterisco.

É válido que o asterisco ocorra na primeira posição de caractere, nesse caso, o nome consiste exclusivamente nos caracteres gerados pelo gerenciador de filas.

Este é um campo de entrada. O comprimento deste campo é fornecido por *LNQN* (*LNQN*). O valor inicial desse campo é 'AMQ.\*', preenchido com espaços em branco.

**ODIDC (inteiro assinado de 10 dígitos)**

Número de filas que falharam ao serem abertas

Este é o número de filas na lista de distribuição que não foram abertas com êxito. Se estiver presente, esse campo também será configurado ao abrir uma única fila que não esteja em uma lista de distribuição

**Nota:** Se presente, este campo será configurado somente se o parâmetro **CMPCOD** na chamada *MQOPEN* ou *MQPUT1* for *CCOK* ou *CCWARN*; ele não será configurado se o parâmetro **CMPCOD** for *CCFAIL*.

Esse é um campo de saída. O valor inicial desse campo é 0. Esse campo será ignorado se *ODVER* for menor que *ODVER2*.

### **ODKDC (número inteiro assinado de 10 dígitos)**

Número de filas locais abertas com êxito..

Este é o número de filas na lista de distribuição que são resolvidas para as filas locais e que foram abertas com sucesso. A contagem não inclui filas que são resolvidas para filas remotas (mesmo que uma fila de transmissão local seja usada inicialmente para armazenar a mensagem). Se estiver presente, esse campo também será configurado ao abrir uma única fila que não esteja em uma lista de distribuição

Esse é um campo de saída. O valor inicial desse campo é 0. Esse campo será ignorado se *ODVER* for menor que *ODVER2*.

### **ODMN (sequência de caracteres de 48 bytes).**

Nome do gerenciador de fila de objetos

Este é o nome do gerenciador de filas no qual o objeto *ODON* está definido. Os caracteres que são válidos no nome são os mesmos para *ODON* (consulte anteriormente). Um nome que está inteiramente em branco até o primeiro caractere nulo ou o final do campo denota o gerenciador de fila ao qual o aplicativo está conectado (o gerenciador de filas locais)...

Os seguintes pontos aplicam-se aos tipos de objeto indicados:

- Se *ODOT* for *OTTOP*, *OTNLST*, *OTPRO* ou *OTQM*, *ODMN* deverá estar em branco ou ser o nome do gerenciador de fila local
- Se *ODON* for o nome de uma fila modelo, o gerenciador de filas criará uma fila dinâmica com os atributos da fila modelo e retornará no campo *ODMN* o nome do gerenciador de filas no qual a fila é criada; este é o nome do gerenciador de fila local. Uma fila modelo pode ser especificada apenas na chamada *MQOPEN*; uma fila modelo não é válida na chamadas *MQPUT1*.
- Se *ODON* for o nome de uma fila de clusters e *ODMN* estiver em branco, o destino real de mensagens enviadas usando o identificador de filas retornado pela chamada *MQOPEN* será escolhido pelo gerenciador de filas (ou saída de carga de trabalho do cluster, se um estiver instalado) conforme a seguir:
  - Se o *OOBND0* for especificado, o gerenciador de filas selecionará uma instância da fila de clusters durante o processamento da chamada *MQOPEN* e todas as mensagens colocadas usando essa manipulação de filas serão enviadas para essa instância
  - Se *OOBNDN* for especificado, o gerenciador de fila poderá escolher uma instância diferente da fila de destino (residindo em um gerenciador de filas diferente no cluster) para cada chamada *MQPUT* sucessiva que usa essa manipulação de fila.

Se o aplicativo precisar enviar uma mensagem para uma instância *específica* de uma fila de clusters (ou seja, uma instância de filas que reside em um determinado gerenciador de fila no cluster), o aplicativo deverá especificar o nome desse gerenciador de filas no campo *ODMN*. Isso força o gerenciador de fila local a enviar a mensagem para o gerenciador de fila de destino especificado

- Se o objeto que está sendo aberto for uma lista de distribuição (ou seja, *ODREC* for maior que zero), *ODMN* deverá estar em branco ou a sequência nula. Se essa condição não for atendida, a chamada falhará com o código de razão *RC2153*

Este é um campo de entrada / saída para a chamada *MQOPEN* quando *ODON* é o nome de uma fila modelo e um campo somente de entrada em todos os outros casos. O comprimento deste campo é fornecido por *LNQMN*. O valor inicial desse campo é 48 caracteres em branco.

### **ODON (sequência de caracteres de 48 bytes).**

Nome do objeto.

Este é o nome local do objeto, conforme definido no gerenciador de filas identificado por *ODMN*. O nome pode conter os seguintes caracteres:

- Caracteres alfabéticos maiúsculos (A-Z)
- Caracteres alfabéticos minúsculos (a-z)

- Dígitos numéricos (0 a 9)
- Ponto (.), barra (/), sublinhado (\_), porcentagem (%)

O nome não deve conter espaços em branco iniciais ou integrados, mas pode conter espaços em branco finais. Um caractere nulo pode ser usado para indicar o término de dados significativos no nome; o nulo e quaisquer caracteres subsequentes são tratados como espaços em branco. As restrições a seguir aplicam-se aos ambientes indicados:

- Em sistemas que usam EBCDIC Katakana, caracteres minúsculos não podem ser usados.
- No IBM i, os nomes contendo caracteres minúsculos, barra ou porcentagem devem ser colocados entre aspas quando especificados em comandos. Essas aspas não devem ser especificadas para nomes que ocorrem como campos em estruturas ou como parâmetros em chamadas..

Os seguintes pontos aplicam-se aos tipos de objeto indicados:

- Se *ODON* for o nome de uma fila modelo, o gerenciador de filas criará uma fila dinâmica com os atributos da fila modelo e retornará no campo *ODON* o nome da fila criada. Uma fila modelo pode ser especificada apenas na chamada *MQOPEN*; uma fila modelo não é válida na chamadas *MQPUT1* .
- Se o objeto que está sendo aberto for uma lista de distribuição (ou seja, *ODREC* está presente e maior que zero), *ODON* deverá estar em branco ou a sequência nula. Se essa condição não for atendida, a chamada falhará com o código de razão RC2152
- Se *ODOT* for *OTQM*, regras especiais se aplicarão; nesse caso, o nome deverá estar inteiramente em branco até o primeiro caractere nulo ou o final do campo...
- Se *ODON* for o nome de uma fila de alias com *TARGETYPE (TOPIC)*, uma verificação de segurança será feita primeiro na fila de alias nomeada, como é normal para o uso de filas de alias. Se essa verificação de segurança for bem-sucedida, esta chamada *MQOPEN* continuará e se comportará como um *MQOPEN* de um *OTTOP*, incluindo fazer uma verificação de segurança com relação ao objeto do tópico administrativo.

Este é um campo de entrada / saída para a chamada *MQOPEN* quando *ODON* é o nome de uma fila modelo e um campo somente de entrada em todos os outros casos O comprimento deste campo é fornecido por *LNQN (LNQN)*. O valor inicial desse campo é 48 caracteres em branco.

O nome completo do tópico pode ser construído de dois campos diferentes: *ODON* e *ODOS*. Para obter detalhes de como esses dois campos são utilizados, consulte [Combinação de sequências de tópicos](#)

### **ODORO (número inteiro assinado de 10 dígitos)**

Deslocamento do primeiro registro de objeto do início do *MQOD*.

Este é o deslocamento em bytes do primeiro registro de objeto *MQOR* do início da estrutura *MQOD*. O deslocamento pode ser positivo ou negativo *ODORO* é usado apenas quando uma lista de distribuições está sendo aberta O campo será ignorado se *ODREC* for zero.

Quando uma lista de distribuição está sendo aberta, uma matriz de um ou mais registros de objeto *MQOR* deve ser fornecida para especificar os nomes das fila de destino na lista de distribuição. Isso pode ser feito de uma de duas maneiras:

- Usando o campo de deslocamento *ODORO*

Nesse caso, o aplicativo deve declarar sua própria estrutura contendo um *MQOD* seguido pela matriz de registros *MQOR* (com quantos elementos de matriz forem necessários) e configurar *ODORO* para o deslocamento do primeiro elemento na matriz a partir do início do *MQOD*. Deve-se ter cuidado para assegurar que esse deslocamento esteja correto

- Usando o campo de ponteiro *ODORP* .

Neste caso, o aplicativo pode declarar a matriz de estruturas *MQOR* separadamente da estrutura *MQOD* e configurar *ODORP* para o endereço da matriz.

Qualquer que seja a técnica escolhida, um de *ODORO* e *ODORP* deve ser usado; a chamada falha com o código de razão RC2155 se ambos forem zero ou ambos forem diferentes de zero.



Este é um campo de entrada. O valor inicial desse campo é 0. Esse campo será ignorado se *ODVER* for menor que *ODVER2*.

### **ODORP (ponteiro)**

Endereço do primeiro registro de objeto..

Este é o endereço do primeiro registro de objeto MÇOR.. *ODORP* é usado apenas quando uma lista de distribuições está sendo aberta O campo será ignorado se *ODREC* for zero.

Este é um campo de entrada. O valor inicial deste campo é o indicador nulo. *ODORP* ou *ODORO* pode ser usado para especificar os registros de objetos, mas não ambos; consulte a descrição do campo *ODORO* anteriormente para obter detalhes.. Se *ODORP* não for usado, ele deverá ser configurado para o ponteiro nulo ou bytes nulos. Esse campo será ignorado se *ODVER* for menor que *ODVER2*.

### **ODOS (MÇCHARV)**

O *ODOS* especifica o nome longo do objeto a ser usado

Esse campo é referenciado apenas para determinados valores de *ODOT*. Consulte a descrição de [ODOT](#) para obter detalhes de quais valores indicam que esse campo é utilizado

Se *ODOS* for especificado incorretamente, de acordo com a descrição de como usar a estrutura [MÇCHARV](#) ou se exceder o comprimento máximo, a chamada falhará com o código de razão RC2441.

Este é um campo de entrada. Os valores iniciais dos campos nessa estrutura são os mesmos da estrutura MÇCHARV.

O nome completo do tópico pode ser construído de dois campos diferentes: *ODON* e *ODOS*. Para obter detalhes de como esses dois campos são utilizados, consulte [Combinação de sequências de tópicos](#) Esse campo será ignorado se *ODVER* for menor que *ODVER4*..

### **ODOT (número inteiro assinado de 10 dígitos)**

Tipo de objeto.

Tipo de objeto sendo nomeado em *ODON*. Os valores possíveis são:

#### **OTQ**

Fila. O nome do objeto está localizado em *ODON*

#### **OTNLST**

Lista de nomes O nome do objeto está localizado em *ODON*

#### **OTPRO**

process definition. O nome do objeto está localizado em *ODON*

#### **OTQM**

Gerenciador de Filas O nome do objeto está localizado em *ODON*

#### **OTTOP**

:NONE. O nome completo do tópico pode ser construído de dois campos diferentes: *ODON* e *ODOS*.

Para obter detalhes sobre como esses dois campos são utilizados, consulte [Combinando sequências de tópicos](#).

Se o objeto identificado pelo campo *ODON* não puder ser localizado, a chamada falhará com o código de razão RC2425 , mesmo se houver uma cadeia especificada em *ODOS*

Isto é sempre um campo de entrada. O valor inicial deste campo é OTQ.

### **ODREC (número inteiro assinado de 10 dígitos)**

Número de registros de objeto presentes..

Este é o número de registros de objeto MÇOR que foram fornecidos pelo aplicativo Se esse número for maior que zero, indica que uma lista de distribuição está sendo aberta, com *ODREC* sendo o número de filas de destino na lista. É válido para uma lista de distribuição conter apenas um destino.

O valor de *ODREC* não deve ser menor que zero e se for maior que zero *ODOT* deve ser OTQ; a chamada falhará com o código de razão RC2154 se essas condições não forem satisfeitas.

Este é um campo de entrada. O valor inicial desse campo é 0. Esse campo será ignorado se *ODVER* for menor que *ODVER2*.

### **ODRMN (sequência de caracteres de 48 bytes)**

Nome do gerenciador de filas resolvido.

Este é o nome do gerenciador de fila de destino após a resolução do nome ter sido executada pelo gerenciador de fila local. O nome retornado é o nome do gerenciador de filas que possui a fila identificada por *ODRQN*. *ODRMN* pode ser o nome do gerenciador de fila local..

Se *ODRQN* for uma fila compartilhada pertencente ao grupo de filas compartilhadas ao qual o gerenciador de filas locais pertence, *ODRMN* será o nome do grupo de filas compartilhadas. Se a fila for de propriedade de algum outro grupo de filas compartilhadas, o *ODRQN* poderá ser o nome do grupo de filas compartilhadas ou o nome de um gerenciador de filas que é um membro do grupo de filas compartilhadas (a natureza do valor retornado é determinada pelas definições de fila que existem no gerenciador de fila local).

Um valor não em branco será retornado apenas se o objeto for uma única fila aberta para procura, entrada ou saída (ou qualquer combinação). Se o objeto aberto for qualquer um dos seguintes, *ODRMN* será configurado como espaços em branco:

- Não uma fila
- Uma fila, mas não aberta para procura, entrada ou saída
- Uma fila de clusters com *OOBNDN* especificado (ou com *OOBNDQ* em vigor quando o atributo da fila **DefBind** tiver o valor *BNDNOT*)
- Uma lista de distribuição

Esse é um campo de saída. O comprimento deste campo é fornecido por *LNQN* (*LNQN*). O valor inicial deste campo é a cadeia nula em C e 48 caracteres em branco em outras linguagens de programação. Este campo será ignorado se *ODVER* for menor que *ODVER3*..

### **ODRO (MQCHARV)**

*ODRO* é o nome do objeto longo depois que o gerenciador de filas resolve o nome fornecido em *ODON*

Esse campo é retornado apenas para determinados tipos de objetos, tópicos e aliases de filas que fazem referência a um objeto de tópico

Se o nome do objeto longo for fornecido em *ODOS* e nada for fornecido em *ODON*, o valor retornado nesse campo será igual ao fornecido em *ODOS*.

Se esse campo for omitido (ou seja, *ODRO.VSBufSize* é zero), o *ODRO* não é retornado, mas o comprimento é retornado no *ODRO.VSLength*. Se o comprimento for menor que o *ODRO* completo, ele será truncado e retornará quantos caracteres mais à direita couber no comprimento fornecido.

Se *ODRO* for especificado incorretamente, de acordo com a descrição de como usar a estrutura *MQCHARV* ou se exceder o comprimento máximo, a chamada falhará com o código de razão *RC2520*. Esse campo será ignorado se *ODVER* for menor que *ODVER4*..

### **ODRQN (cadeia de caracteres de 48 bytes).**

Nome da fila resolvido.

Este é o nome da fila de destino após a resolução do nome ter sido executada pelo gerenciador de fila local. O nome retornado é o nome de uma fila existente no gerenciador de filas identificado por *ODRMN*.

Um valor não em branco será retornado apenas se o objeto for uma única fila aberta para procura, entrada ou saída (ou qualquer combinação). Se o objeto aberto for qualquer um dos seguintes, *ODRQN* será configurado como espaços em branco:

- Não uma fila
- Uma fila, mas não aberta para procura, entrada ou saída
- Uma lista de distribuição

- Uma fila de alias que faz referência a um objeto de tópico (consulte [“ODRO \(MÇCHARV\)”](#) na página 1206 no lugar)

Esse é um campo de saída. O comprimento deste campo é fornecido por LNQN (LNQN). O valor inicial deste campo é a cadeia nula em C e 48 caracteres em branco em outras linguagens de programação. Este campo será ignorado se *ODVER* for menor que *ODVER3*..

### **ODRRO (número inteiro assinado de 10 dígitos)**

Deslocamento do primeiro registro de resposta do início do MÇOD.

Este é o deslocamento em bytes do primeiro registro de resposta MÇRR do início da estrutura MÇOD. O deslocamento pode ser positivo ou negativo *ODRRO* é usado apenas quando uma lista de distribuições está sendo aberta O campo será ignorado se *ODREC* for zero.

Quando uma lista de distribuição está sendo aberta, uma matriz de um ou mais registros de resposta MÇRR pode ser fornecida para identificar as filas que falharam ao abrir (campo *RRCC* no MÇRR) e a razão para cada falha (campo *RRREA* no MÇRR)... Os dados são retornados na matriz de registros de resposta na mesma ordem em que os nomes de filas ocorrem na matriz de registros de objetos O gerenciador de filas configura os registros de resposta somente quando o resultado da chamada é misto (ou seja, algumas filas foram abertas com êxito enquanto outras falharam, ou todas falharam, mas por razões diferentes); o código de razão RC2136 da chamada indica esse caso.. Se o mesmo código de razão se aplicar a todas as filas, esse motivo será retornado no parâmetro **REASON** da chamada MÇOPEN ou MÇPUT1 e os registros de resposta não serão configurados. Os registros de resposta são opcionais, mas se forem fornecidos, deverá haver *ODREC* deles.

Os registros de resposta podem ser fornecidos da mesma forma que os registros do objeto, especificando um deslocamento em *ODRRO* ou especificando um endereço em *ODRRP* ; Consulte a descrição do *ODORO* anteriormente para obter detalhes de como fazer isso No entanto, não mais de um de *ODRRO* e *ODRRP* pode ser usado; a chamada falhará com o código de razão RC2156 se ambos forem diferentes de zero

Para a chamada MÇPUT1 , esses registros de resposta são usados para retornar informações sobre erros que ocorrem quando a mensagem é enviada para as filas na lista de distribuição, bem como erros que ocorrem quando as filas são abertas. O código de conclusão e o código de razão da operação put para uma fila substituem aqueles da operação open para essa fila apenas se o código de conclusão do último era CCOK ou CCWARN

Este é um campo de entrada. O valor inicial desse campo é 0. Esse campo será ignorado se *ODVER* for menor que *ODVER2*.

### **ODRRP (ponteiro)**

Endereço do primeiro registro de resposta..

Este é o endereço do primeiro registro de resposta do MÇRR *ODRRP* é usado apenas quando uma lista de distribuições está sendo aberta O campo será ignorado se *ODREC* for zero.

*ODRRP* ou *ODRRO* pode ser usado para especificar os registros de resposta, mas não ambos; consulte a descrição anterior do campo *ODRRO* para obter detalhes Se *ODRRP* não for usado, ele deverá ser configurado para o ponteiro nulo ou bytes nulos.

Este é um campo de entrada. O valor inicial deste campo é o indicador nulo. Esse campo será ignorado se *ODVER* for menor que *ODVER2*.

### **ODSID (sequência de caracteres de 4 bytes)**

Identificador de estruturação.

O valor deve ser:.

#### **ODSIDV**

Identificador para a estrutura do descritor de objetos

Isto é sempre um campo de entrada. O valor inicial desse campo é *ODSIDV*.

## ODSS (MÇCHARV)

ODSS contém a sequência usada para fornecer os critérios de seleção usados ao recuperar mensagens de uma fila.

ODSS não deve ser fornecido nos seguintes casos:

- Se *ODOT* não for OTQ
- Se a fila que está sendo aberta não estiver sendo aberta usando uma das opções de entrada, OOINP\*

Se *ODSS* for fornecido nesses casos, a chamada falhará com código de razão RC2516.

Se *ODSS* for especificado incorretamente, de acordo com a descrição de como usar a estrutura MÇCHARV ou se exceder o comprimento máximo, a chamada falhará com o código de razão RC2519. Esse campo será ignorado se *ODVER* for menor que ODVER4..

## ODUDC (número inteiro assinado de 10 dígitos)

Número de filas remotas abertas com êxito

Este é o número de filas na lista de distribuição que são resolvidas para filas remotas e que foram abertas com êxito. Se estiver presente, esse campo também será configurado ao abrir uma única fila que não esteja em uma lista de distribuição.

Esse é um campo de saída. O valor inicial desse campo é 0. Esse campo será ignorado se *ODVER* for menor que ODVER2.

## ODVER (número inteiro assinado de 10 dígitos)

Número de versão da estrutura.

O valor deve ser um dos seguintes:

### ODVER1

Estrutura do descritor de objeto Version-1

### ODVER2

Estrutura do descritor de objeto Version-2 .

### ODVER3

Estrutura do descritor de objeto Version-3

### ODVER4

Estrutura do descritor de objeto Version-4

Os campos que existem apenas nas versões mais recentes da estrutura são identificados como tal nas descrições dos campos. A constante a seguir especifica o número da versão atual:

### ODVERC

A versão atual da estrutura do descritor de objetos

Isto é sempre um campo de entrada. O valor inicial desse campo é ODVER1.

## Valores iniciais

Nome de Campo	Nome da constante	Valor da constante
<i>ODSID</i>	ODSIDV	'OD--'
<i>ODVER</i>	ODVER1	1
<i>ODOT</i>	OTQ	1
<i>ODON</i>	Nenhum	Espaços em branco
<i>ODMN</i>	Nenhum	Espaços em branco

Tabela 713. Campos em MQOD (continuação)

Nome de Campo	Nome da constante	Valor da constante
ODDN	Nenhum	'AMQ.*'
ODAU	Nenhum	Espaços em branco
ODREC	Nenhum	0
ODKDC	Nenhum	0
ODUDC	Nenhum	0
ODIDC	Nenhum	0
ODORO	Nenhum	0
ODRRO	Nenhum	0
ODORP	Nenhum	Ponteiro nulo ou bytes nulos
ODRRP	Nenhum	Ponteiro nulo ou bytes nulos
ODASI	SINONE	Nulos
ODRQN	Nenhum	Espaços em branco
ODRMN	Nenhum	Espaços em branco
ODOS	Conforme definido para MQCHARV	Conforme definido para MQCHARV
ODRO	Conforme fornecido em ODOS	Conforme fornecido em ODOS
ODSS	Nenhum	Espaços em branco
<b>Notas:</b>		
1. O símbolo ~ representa um único caractere em branco.		

## Declaração RPG

```

D*..1.....2.....3.....4.....5.....6.....7..
D*
D* MQOD Structure
D*
D*
D* Structure identifier
D  ODSID          1      4    INZ('OD ')
D*
D* Structure version number
D  ODVER          5      8I 0 INZ(1)
D*
D* Object type
D  ODOT           9     12I 0 INZ(1)
D*
D* Object name
D  ODON          13     60    INZ
D*
D* Object queue manager name
D  ODMN          61     108   INZ
D*
D* Dynamic queue name
D  ODDN         109     156   INZ('AMQ.*')
D*
D* Alternate user identifier
D  ODAU         157     168   INZ
D*
** Number of object records
D* present
D  ODREC        169     172I 0 INZ(0)

```

```

D*
** Number of local queues opened
D* successfully
D ODKDC          173    176I 0 INZ(0)
D*
** Number of remote queues opened
D* successfully
D ODUDC          177    180I 0 INZ(0)
D*
** Number of queues that failed to
D* open
D ODIDC          181    184I 0 INZ(0)
D*
** Offset of first object record
D* from start of MQOD
D ODORO          185    188I 0 INZ(0)
D*
** Offset of first response record
D* from start of MQOD
D ODRRO          189    192I 0 INZ(0)
D*
D* Address of first object record
D ODORP          193    208*  INZ(*NULL)
D*
** Address of first response
D* record
D ODRRP          209    224*  INZ(*NULL)
D*
D* Alternate security identifier
D ODASI          225    264    INZ(X'0000000000000000-
D                000000000000000000000000-
D                000000000000000000000000-
D                000000000000')
D*
D* Resolved queue name
D ODRQN          265    312    INZ
D*
D* Resolved queue manager name
D ODRMN          313    360    INZ
D*
D* reserved field
D ODRE1          361    364I 0 INZ(0)
D*
D* reserved field
D ODRS2          365    368I 0 INZ(0)
D*
D* Object long name
D* Address of variable length string
D ODOSCHRP       369    384*  INZ(*NULL)
D* Offset of variable length string
D ODOSCHRO       385    388I 0 INZ(0)
D* Size of buffer
D ODOSVSBS       389    392I 0 INZ(-1)
D* Length of variable length string
D ODOSCHRL       393    396I 0 INZ(0)
D* CCSID of variable length string
D ODOSCHRC       397    400I 0 INZ(-3)
D*
D* Message Selector
D* Address of variable length string
D ODSSCHRP       401    416*  INZ(*NULL)
D* Offset of variable length string
D ODSSCHRO       417    420I 0 INZ(0)
D* Size of buffer
D ODSSVSBS       421    424I 0 INZ(-1)
D* Length of variable length string
D ODSSCHRL       425    428I 0 INZ(0)
D* CCSID of variable length string
D ODSSCHRC       429    432I 0 INZ(-3)
D*
D* Resolved long object name
D* Address of variable length string
D ODRSOCHRP      433    448*  INZ(*NULL)
D* Offset of variable length string
D ODRSOCHRO      449    452I 0 INZ(0)
D* Size of buffer
D ODRSOVSBS      453    456I 0 INZ(-1)
D* Length of variable length string
D ODRSOCHRL      457    460I 0 INZ(0)
D* CCSID of variable length string
D ODRSOCHRC      461    464I 0 INZ(-3)
D*

```

D\* Alias queue resolved object type  
D ODRT 465 468I 0 INZ(0)

## IBM i MQOR (Object record) em IBM i

A estrutura MQOR é usada para especificar o nome da fila e o nome do gerenciador de filas de uma única fila de destino

### Visão Geral

**Propósito:** MQOR é uma estrutura de entrada para as chamadas MQOPEN e MQPUT1

**Conjunto de caracteres e codificação:** os dados em MQOR devem estar no conjunto de caracteres fornecido pelo atributo do gerenciador de filas do **CodedCharSetId** e na codificação do gerenciador de fila local fornecido por ENNAT.. No entanto, se o aplicativo estiver em execução como um cliente IBM MQ, a estrutura deverá estar no conjunto de caracteres e na codificação do cliente..

**Uso:** fornecendo uma matriz dessas estruturas na chamada MQOPEN, é possível abrir uma lista de filas; essa lista é chamada *lista de distribuição*. Cada mensagem colocada usando o identificador de filas retornado por essa chamada MQOPEN é colocada em cada uma das filas na lista, se a fila tiver sido aberta com êxito

- [“Campos” na página 1211](#)
- [“Valores iniciais” na página 1211](#)
- [“Declaração RPG” na página 1211](#)

### Campos

A estrutura MQOR contém os seguintes campos; os campos são descritos em **ordem alfabética**:

#### ORMN (sequência de caracteres de 48 bytes)

Nome do gerenciador de fila de objetos

Este é o mesmo que o campo *ODMN* na estrutura MQOD (consulte MQOD para obter detalhes)

Isto é sempre um campo de entrada. O valor inicial desse campo é 48 caracteres em branco.

#### ORON (cadeia de caracteres de 48 bytes),

Nome do objeto.

É o mesmo que o campo *ODON* na estrutura MQOD (consulte MQOD para obter detalhes), exceto que:

- Ele deve ser o nome de uma fila
- Não deve ser o nome de uma fila modelo.

Isto é sempre um campo de entrada. O valor inicial desse campo é 48 caracteres em branco.

### Valores iniciais

Tabela 714. Campos em MQOR		
Nome de Campo	Nome da constante	Valor da constante
ORON	Nenhum	Espaços em branco
ORMN	Nenhum	Espaços em branco

### Declaração RPG

D\* .1.....2.....3.....4.....5.....6.....7..  
D\*

D*	MQOR Structure			
D*				
D*	Object name			
D	ORON	1	48	INZ
D*	Object queue manager name			
D	ORMN	49	96	INZ

## MQPD-Descrição de propriedades

O **MQPD** é usado para definir os atributos de uma propriedade

### Visão Geral

**Propósito:** a estrutura é um parâmetro de entrada / saída na chamada MQSETMP e um parâmetro de saída na chamada MQINQMP.

**Conjunto de caracteres e codificação:** os dados em MQPD devem estar no conjunto de caracteres do aplicativo e codificação do aplicativo (ENNAT).

- [“Campos” na página 1212](#)
- [“Valores iniciais” na página 1214](#)
- [“Declaração RPG” na página 1215](#)

### Campos

A estrutura MQPD contém os campos a seguir; os campos são descritos em **ordem alfabética**:

#### PDCT (número inteiro assinado de 10 dígitos)

Isto descreve a qual contexto de mensagem a propriedade pertence.

Quando um gerenciador de filas recebe uma mensagem contendo uma propriedade definida por IBM MQ que o gerenciador de fila reconhece como sendo incorreta O gerenciador de filas corrige o valor do campo *PDCT*.

A seguinte opção pode ser especificada:

##### PDUSC

A propriedade é associada com o contexto do usuário.

Não é necessária nenhuma autorização especial para poder definir uma propriedade associada ao contexto do usuário utilizando a chamada MQSETMP.

Se a opção descrita anteriormente não for necessária, a seguinte opção poderá ser usada:

##### PDNOC

A propriedade não é associada com um contexto de mensagem.

Um valor não reconhecido é rejeitado com um código *PDREA* de RC2482.

Este é um campo de entrada / saída para a chamada MQSETMP e um campo de saída da chamada MQINQMP. O valor inicial deste campo é PDNOC..

#### PDOPYOPT (número inteiro assinado de 10 dígitos)

Isto descreve em qual tipo de mensagens a propriedade deve ser copiada.

Este é um campo de saída somente para propriedades IBM MQ definidas reconhecidas; IBM MQ configura o valor apropriado.

Quando um gerenciador de filas recebe uma mensagem contendo uma propriedade definida por IBM MQ que o gerenciador de fila reconhece como sendo incorreta O gerenciador de filas corrige o valor do campo *CopyOptions*.

É possível especificar uma ou mais dessas opções Para especificar mais de uma opção, inclua os valores juntos (não inclua a mesma constante mais de uma vez) ou combine os valores usando a operação OR bit a bit (se a linguagem de programação suportar operações de bit).



**COPFOR**

Esta propriedade é copiada para uma mensagem que está sendo encaminhada

**COPPUB**

Esta propriedade é copiada na mensagem recebida por um assinante quando uma mensagem está sendo publicada.

**COPREP**

Esta propriedade é copiada em uma mensagem de resposta.

**COPRP**

Esta propriedade é copiada em uma mensagem de relatório.

**COPALL**

Esta propriedade é copiada em todos os tipos de mensagens subsequentes.

**COPNON**

Esta propriedade não é copiada em uma mensagem

**Opção padrão:** A seguinte opção pode ser especificada para fornecer o conjunto padrão de opções de cópia:

**COPDEF**

Essa propriedade é copiada em uma mensagem sendo encaminhada, em uma mensagem de relatório ou em uma mensagem recebida por um assinante quando uma mensagem está sendo publicada.

Isso é equivalente a especificar a combinação de opções COPFOR, mais COPRP, mais COPPUB.

Se nenhuma das opções descritas anteriormente forem necessárias, use a seguinte opção:

**COPNON**

Use este valor para indicar que nenhuma outra opção de cópia tenha sido especificada; programaticamente, não existe nenhum relacionamento entre esta propriedade e mensagens subsequentes. Isto é sempre retornado para propriedades do descritor de mensagens.

Este é um campo de entrada / saída para a chamada MQSETMP e um campo de saída da chamada MQINQMP. O valor inicial desse campo é COPDEF.

**PDOPT (número inteiro assinado de 10 dígitos)**

O valor deve ser:.

**PDNONE**

Nenhuma opção especificada

Isto é sempre um campo de entrada. O valor inicial deste campo é PDNONE.

**PDSID (número inteiro assinado de 10 dígitos).**

Este é o identificador de estrutura; o valor deve ser:

**PSIDV**

Identificador para a estrutura do descritor de propriedades

Isto é sempre um campo de entrada. O valor inicial desse campo é **PSIDV**.

**PDSUP (número inteiro assinado de 10 dígitos)**

Este campo descreve qual nível de suporte para a propriedade de mensagens é necessário para o gerenciador de filas, para que a mensagem que contém esta propriedade seja colocada em uma fila. Isso se aplica apenas às propriedades definidas pelo IBM MQ; o suporte para todas as outras propriedades é opcional

O campo é configurado automaticamente para o valor correto quando a propriedade definida por IBM MQ é conhecida pelo gerenciador de filas. Se a propriedade não for reconhecida, o PDSUP será designado. Quando um gerenciador de filas recebe uma mensagem contendo uma propriedade definida por IBM MQ que o gerenciador de fila reconhece como sendo incorreta, o gerenciador de filas corrige o valor do campo *PDSUP*.

Ao configurar uma propriedade definida pelo IBM MQ usando a chamada MQSETMP em um identificador de mensagem em que a opção CMNOVA foi configurada, o PDSUP se torna um campo de entrada. Isso permite que um aplicativo coloque uma propriedade definida pelo IBM MQ, com o valor correto, em que a propriedade não é suportada pelo gerenciador de filas conectado, mas em que a mensagem deve ser processada em outro gerenciador de filas.

O valor PDSUPO é sempre designado para propriedades que não são propriedades definidas pelo IBM MQ..

Um dos valores a seguir é retornado pela chamada MQINQMP ou um dos valores pode ser especificado ao usar a chamada MQSETMP em uma manipulação de mensagem na qual a opção CMNOVA é configurada:

#### **PDSUPO**

A propriedade é aceita por um gerenciador de filas mesmo se ela não for suportada. A propriedade pode ser descartada para a mensagem fluir para um gerenciador de filas que não suporta propriedades de mensagens. Esse valor também é designado às propriedades que não são definidas por IBM MQ.

#### **PDSUPR**

É necessário suporte para a propriedade. A mensagem é rejeitada por um gerenciador de filas que não suporta a propriedade definida pelo IBM MQ. A chamada MQPUT ou MQPUT1 falha com código de conclusão CCFAIL e código de razão RC2490.

#### **PDSUPL**

A mensagem será rejeitada por um gerenciador de filas que não suporta a propriedade definida por IBM MQ se a mensagem for destinada para uma fila local. A chamada MQPUT ou MQPUT1 falha com código de conclusão CCFAIL e código de razão RC2490.

A chamada MQPUT ou MQPUT1 é bem-sucedida se a mensagem é destinada para um gerenciador de filas remotas.

Este é um campo de saída na chamada MQINQMP e um campo de entrada na chamada MQSETMP se a manipulação de mensagem foi criada com a opção CMNOVA configurada. O valor inicial deste campo é PDSUPO.

### **PDVER (número inteiro assinado de 10 dígitos)**

Este é o número da versão da estrutura; o valor deve ser:

#### **PDVER1**

Estrutura do descritor da propriedade versão 1.

A constante a seguir especifica o número da versão atual:

#### **PDVERC**

Versão atual da estrutura do descritor da propriedade.

Isto é sempre um campo de entrada. O valor inicial desse campo é **PDVER1**.

### **Valores iniciais**

<i>Tabela 715. Campos em MQPD</i>		
<b>Nome de Campo</b>	<b>Nome da constante</b>	<b>Valor da constante</b>
<i>PDSID</i>	PDSIDV	'PD'
<i>PDVER</i>	PDVER1	1
<i>PDOPT</i>	PDNONE	0
<i>PDSUP</i>	PDSUPO	0
<i>PDCT</i>	PDNOC	0

Tabela 715. Campos em MQPD (continuação)

Nome de Campo	Nome da constante	Valor da constante
PDCPYOPT	COPDEF	0

## Declaração RPG

```

D* MQDMHO Structure
D*
D*
D* Structure identifier
D DMSID          1      4   INZ('DMHO')
D*
D* Structure version number
D DMVER          5      8I 0 INZ(1)
D*
D* Options that control the action of MQDLTMH
D DMOPT          9     12I 0 INZ(0)
    
```



## MQPMO (opções Put-message) em IBM i

A estrutura MQPMO permite que o aplicativo especifique opções que controlem como as mensagens são colocadas em filas ou publicadas em tópicos

### Visão Geral

#### Propósito

A estrutura é um parâmetro de entrada / saída nas chamadas MQPUT e MQPUT1 .

#### Versão

A versão atual do MQPMO é PMVER2. Os campos que existem apenas nas versões mais recentes da estrutura são identificados como tal nas descrições a seguir:

O arquivo COPY fornecido contém a versão mais recente do MQPMO que é suportada pelo ambiente, mas com o valor inicial do campo *PMVER* configurado como PMVER1... Para usar campos que não estão presentes na estrutura version-1 , o aplicativo deve configurar o campo *PMVER* para o número da versão necessária.

#### Conjunto de caracteres e codificação

Os dados no MQPMO devem estar no conjunto de caracteres fornecido pelo atributo do gerenciador de filas **CodedCharSetId** e pela codificação do gerenciador de fila local fornecido pelo ENNAT No entanto, se o aplicativo estiver em execução como um cliente IBM MQ , a estrutura deverá estar no conjunto de caracteres e na codificação do cliente..

- [“Campos” na página 1215](#)
- [“Valores iniciais” na página 1230](#)
- [“Declaração RPG” na página 1230](#)

### Campos

A estrutura MQPMO contém os campos a seguir; os campos são descritos em ordem alfabética:

#### PMCT (número inteiro assinado de 10 dígitos).

Identificador de objeto da fila de entrada

Se PMPASI ou PMPASA for especificado, este campo deverá conter o identificador de fila de entrada a partir do qual as informações de contexto a serem associadas com a mensagem que está sendo colocada serão obtidas

Se PMPASI e PMPASA não forem especificados, este campo será ignorado

Este é um campo de entrada. O valor inicial desse campo é 0.

### **PMIDC (número inteiro assinado de 10 dígitos)**

O número de mensagens que não puderam ser enviadas

Este é o número de mensagens que não puderam ser enviadas para as filas na lista de distribuição A contagem inclui filas que falharam ao serem abertas e filas que foram abertas com êxito, mas para as quais a operação put falhou. Esse campo também é configurado ao colocar uma mensagem em uma única fila que não está na lista de distribuição.

**Nota:** Esse campo será configurado somente se o parâmetro **CMPCOD** na chamada MQPUT ou MQPUT1 for CCOK ou CCWARN; ele não será configurado se o parâmetro **CMPCOD** for CCFAIL.

Esse é um campo de saída. O valor inicial desse campo é 0. Esse campo não será configurado se *PMVER* for menor que *PMVER2*.

### **PMKDC (número inteiro assinado de 10 dígitos)**

Número de mensagens enviadas com êxito para as filas locais

Este é o número de mensagens que a chamada MQPUT ou MQPUT1 atual enviou com sucesso para filas na lista de distribuição que são filas locais. A contagem não inclui mensagens enviadas para filas que são resolvidas para filas remotas (mesmo que uma fila de transmissão local seja usada inicialmente para armazenar a mensagem). Esse campo também é configurado ao colocar uma mensagem em uma única fila que não está na lista de distribuição.

Esse é um campo de saída. O valor inicial desse campo é 0. Esse campo não será configurado se *PMVER* for menor que *PMVER2*.

### **PMOPT (número inteiro assinado de 10 dígitos).**

Opções que controlam a ação de MQPUT e MQPUT1.

Qualquer um ou nenhum dos seguintes pode ser especificado. Se mais de um for necessário, os valores poderão ser adicionados (não inclua a mesma constante mais de uma vez). As combinações que não são válidas são anotadas; quaisquer outras combinações são válidas

**Opções de publicação:** As opções a seguir controlam a maneira como as mensagens são publicadas em um tópico..

#### **PMSRTO**

Quaisquer informações preenchidas nos campos MDRQ e MDRM do MQMD desta publicação não são transmitidas aos assinantes. Se essa opção for usada com uma opção de relatório que requeira um ReplyToQ, a chamada falhará com RC2027

#### **PMRET**

A publicação que está sendo enviada deve ser retida pelo gerenciador de filas. Isto permite que um assinante solicite uma cópia desta publicação após o momento que ela foi publicada, usando a chamada MQSUBRQ. Também permite que uma publicação seja enviada para aplicações que façam sua assinatura após o momento em que esta publicação foi feita, a menos que optem por não enviá-la usando a opção SONEWP. Se um aplicativo receber uma publicação que foi retida, ela será indicada pela propriedade de mensagem mq.IsRetained dessa publicação.

Somente uma publicação pode ser retida em cada nó da árvore de tópicos. Ou seja, se já houver uma publicação retida para este tópico, publicada por qualquer outro aplicativo, ela será substituída por esta publicação. Portanto, é melhor evitar que mais de um publicador mantenha mensagens no mesmo tópico.

Quando as publicações retidas são solicitadas por um assinante, a assinatura usada pode conter um curinga no tópico, nesse caso um número de publicações retidas pode corresponder (em vários nós na árvore de tópicos) e várias publicações podem ser enviadas para o aplicativo solicitante. Consulte a descrição da chamada [“MQSUBRQ-Pedido de assinatura”](#) na página 824 para obter mais detalhes..

Se essa opção for usada e a publicação não puder ser retida, a mensagem não será publicada e a chamada falhará com RC2479

**Opções de ponto de sincronização:** As opções a seguir estão relacionadas à participação da chamada MQPUT ou MQPUT1 dentro de uma unidade de trabalho:

#### **PMNSYP**

Colocar mensagem com controle de ponto de sincronização.

A solicitação é operar dentro dos protocolos normais de unidade de trabalho. A mensagem não é visível fora da unidade de trabalho até que a unidade de trabalho seja confirmada. Se a unidade de trabalho for recuperada, a mensagem será excluída.

Se essa opção e PMNSYP não forem especificados, o pedido de colocação não estará dentro de uma unidade de trabalho

PMNSYP não deve ser especificado com PMNSYP.

#### **PMNSYP**

Colocar mensagem sem controle de ponto de sincronização.

O pedido é para operar fora dos protocolos normais de unidade de trabalho. A mensagem está disponível imediatamente e não pode ser excluída efetuando backout de uma unidade de trabalho

Se esta opção e PMSYP não forem especificados, o pedido de colocação não estará dentro de uma unidade de trabalho

PMNSYP não deve ser especificado com PMSYP.

**Opções do identificador de mensagens e do identificador de correlação:** As opções a seguir solicitam que o gerenciador de filas gere um novo identificador de mensagens ou identificador de correlação:

#### **PMNMID**

Gerar um novo identificador de mensagem.

Essa opção faz o gerenciador de filas substituir o conteúdo do campo *MDMID* no MQMD por um novo identificador de mensagem. Esse identificador de mensagem é enviado com a mensagem e retornado ao aplicativo na saída da chamada MQPUT ou MQPUT1 .

Essa opção também pode ser especificada quando a mensagem está sendo colocada em uma lista de distribuições; consulte a descrição do campo *PRMID* na estrutura MQPMR para obter detalhes

O uso dessa opção alivia a necessidade de reconfigurar o campo *MDMID* para MINONE antes de cada chamada MQPUT ou MQPUT1 .

#### **PMNCID**

Gerar um novo identificador de correlação.

Essa opção faz com que o gerenciador de filas substitua o conteúdo do campo *MDCID* em MQMD por um novo identificador de correlação. Esse identificador de correlação é enviado com a mensagem e retornado ao aplicativo na saída da chamada MQPUT ou MQPUT1 .

Essa opção também pode ser especificada quando a mensagem está sendo colocada em uma lista de distribuições; consulte a descrição do campo *PRCID* na estrutura MQPMR para obter detalhes

PMNCID é útil em situações em que o aplicativo requer um identificador de correlação exclusivo.

**Opções de grupo e segmento:** A opção a seguir está relacionada ao processamento de mensagens em grupos e segmentos de mensagens lógicas. Essas definições podem ajudar a entender a opção:

#### **Mensagem física**

Esta é a menor unidade de informações que pode ser colocada ou removida de uma fila; geralmente corresponde às informações especificadas ou recuperadas em uma chamada MQPUT, MQPUT1 ou MQGET. Cada mensagem física possui seu próprio descritor de mensagens (MQMD).

Geralmente, as mensagens físicas são distinguidas por valores diferentes para o identificador de mensagens (campo *MDMID* em MQMD), embora isso não seja aplicado pelo gerenciador de filas.

### **Mensagem lógica**

Esta é uma única unidade de informações do aplicativo. Na ausência de restrições do sistema, uma mensagem lógica seria a mesma que uma mensagem física. Mas onde as mensagens lógicas são grandes, as restrições do sistema podem tornar aconselhável ou necessário dividir uma mensagem lógica em duas ou mais mensagens físicas, chamadas *segmentos*.

Uma mensagem lógica que foi segmentada consiste em duas ou mais mensagens físicas que possuem o mesmo identificador de grupo não nulo (campo *MDGID* em MQMD) e o mesmo número de sequência de mensagens (campo *MDSEQ* em MQMD). Os segmentos são diferenciados por valores diferentes para o deslocamento de segmento (campo *MDOFF* em MQMD), que fornece o deslocamento dos dados na mensagem física do início dos dados na mensagem lógica. Como cada segmento é uma mensagem física, os segmentos em uma mensagem lógica geralmente possuem identificadores de mensagem diferentes..

Uma mensagem lógica que não foi segmentada, mas para a qual a segmentação foi permitida pelo aplicativo de envio, também possui um identificador de grupo não nulo, embora neste caso haja apenas uma mensagem física com esse identificador de grupo se a mensagem lógica não pertencer a um grupo de mensagens. Mensagens lógicas para as quais a segmentação foi inibida pelo aplicativo de envio possuem um identificador de grupo nulo (GINONE), a menos que a mensagem lógica pertença a um grupo de mensagens.

### **Grupo de mensagens**

Este é um conjunto de uma ou mais mensagens lógicas que possuem o mesmo identificador de grupo não nulo.. As mensagens lógicas no grupo são distinguidas por valores diferentes para o número de sequência da mensagem, que é um número inteiro no intervalo de 1 a n, em que n é o número de mensagens lógicas no grupo. Se uma ou mais mensagens lógicas forem segmentadas, haverá mais de n mensagens físicas no grupo.

### **PMLOGO**

Mensagens em grupos e segmentos de mensagens lógicas são colocadas em ordem lógica.

Esta opção informa ao gerenciador de filas como o aplicativo coloca mensagens em grupos e segmentos de mensagens lógicas... Ela só pode ser especificada na chamada MQPUT. Ela não é válida na chamada MQPUT1.

Se PMLOGO for especificado, isso indica que o aplicativo usa chamadas MQPUT sucessivas para:

- Colocar os segmentos em cada mensagem lógica na ordem crescente de deslocamento de segmento, iniciando a partir de 0, sem lacunas.
- Coloque todos os segmentos em uma mensagem lógica antes de colocar os segmentos na próxima mensagem lógica
- Colocar as mensagens lógicas em cada grupo de mensagens na ordem crescente de número de sequência da mensagem, iniciando a partir de 1, sem lacunas.
- Coloque todas as mensagens lógicas em um grupo de mensagens antes de colocar as mensagens lógicas no próximo grupo de mensagens

Esta ordem é chamada "ordem lógica".

Como o aplicativo informou ao gerenciador de filas como ele coloca mensagens em grupos e segmentos de mensagens lógicas, o aplicativo não precisa manter e atualizar as informações do grupo e do segmento sobre cada chamada MQPUT, conforme o gerenciador de filas faz isso. Especificamente, isso significa que o aplicativo não precisa configurar os campos *MDGID*, *MDSEQ* e *MDOFF* no MQMD, já que o gerenciador de filas os configura com os valores apropriados. O aplicativo precisa configurar apenas o campo *MDMFL* no MQMD para indicar quando as mensagens pertencem a grupos ou são segmentos de mensagens lógicas e para indicar a última mensagem em um grupo ou último segmento de uma mensagem lógica.

Quando um grupo de mensagens ou mensagem lógica tiver sido iniciado, as chamadas MQPUT subsequentes deverão especificar os sinalizadores MF\* apropriados em *MDMFL* no MQMD. Se o

aplicativo tentar colocar uma mensagem que não está em um grupo quando há um grupo de mensagens não finalizadas, ou colocar uma mensagem que não é um segmento quando há uma mensagem lógica não finalizada, a chamada falhará com o código de razão RC2241 ou RC2242, conforme apropriado. No entanto, o gerenciador de filas retém as informações sobre o grupo de mensagens atual ou a mensagem lógica atual, e o aplicativo pode finalizá-las enviando uma mensagem (possivelmente sem dados de mensagem do aplicativo) especificando MFLMIG ou MFLSEG conforme apropriado, antes de emitir novamente a chamada MQPUT para colocar a mensagem que não está no grupo ou não é um segmento..

Tabela 716 na página 1219 mostra as combinações de opções e sinalizações que são válidas e os valores dos campos *MDGID*, *MDSEQ* e *MDOFF* que o gerenciador de filas usa em cada caso. Combinações de opções e sinalizadores que não são mostrados na tabela não são válidos. As colunas na tabela têm os significados a seguir:

**LOG ORD**

Indica se a opção PMLOGO é especificada na chamada

**MIG**

Indica se a opção MFMIG ou MFLMIG é especificada na chamada

**SEG**

Indica se a opção MFSEG ou MFLSEG é especificada na chamada

**SEG OK**

Indica se a opção MFSEGA é especificada na chamada

**Cur grp**

Indica se um grupo de mensagens atual existe antes da chamada.

**Cur log msg**

Indica se uma mensagem lógica atual existe antes da chamada

**Outras colunas**

Mostram os valores que o gerenciador de filas usa. "Anterior" indica o valor usado para o campo na mensagem anterior para o identificador de filas.

**PMRLOC**

Especifica que o PMRQN na estrutura MQPMO deve ser concluído com o nome da fila local para a qual a mensagem realmente é colocada. O Nome ResolvedQMGré igualmente concluído com o nome do gerenciador de fila local que hospeda a fila local. Veja OORLOQ para saber o que isso significa. Se um usuário estiver autorizado para um put em uma fila, ele terá a autoridade necessária para especificar esse sinalizador na chamada MQPUT. Nenhuma autoridade especial é necessária.

Tabela 716. Opções MQPUT relacionadas às mensagens em grupos e segmentos de mensagens lógicas								
Opções que você especifica				Grupo e status log-msg antes da chamada		Valores que o gerenciador de filas usa		
LOG ORD	MIG	SEG	SEG OK	Cur grp	Cur log msg	MDGID	MDSEQ	MDOFF
Sim	No	No	No	No	No	GINONE	1	0
Sim	No	No	Sim	No	No	Nova ID de grupo	1	0
Sim	No	Sim	Sim ou Não	No	No	Nova ID de grupo	1	0

Tabela 716. Opções MQPUT relacionadas às mensagens em grupos e segmentos de mensagens lógicas (continuação)

Opções que você especifica				Grupo e status log-msg antes da chamada		Valores que o gerenciador de filas usa		
Sim	No	Sim	Sim ou Não	No	Sim	ID de grupo anterior	1	Deslocamento anterior + comprimento do segmento anterior
Sim	Sim	Sim ou Não	Sim ou Não	No	No	Nova ID de grupo	1	0
Sim	Sim	Sim ou Não	Sim ou Não	Sim	No	ID de grupo anterior	Número da sequência anterior + 1	0
Sim	Sim	Sim	Sim ou Não	Sim	Sim	ID de grupo anterior	Número de sequência anterior	Deslocamento anterior + comprimento do segmento anterior
No	No	No	No	Sim ou Não	Sim ou Não	GINONE	1	0
No	No	No	Sim	Sim ou Não	Sim ou Não	Novo ID do grupo se GINONE, caso contrário, valor no campo	1	0
No	No	Sim	Sim ou Não	Sim ou Não	Sim ou Não	Novo ID do grupo se GINONE, caso contrário, valor no campo	1	Valor no campo
No	Sim	No	Sim ou Não	Sim ou Não	Sim ou Não	Novo ID do grupo se GINONE, caso contrário, valor no campo	Valor no campo	0
No	Sim	Sim	Sim ou Não	Sim ou Não	Sim ou Não	Novo ID do grupo se GINONE, caso contrário, valor no campo	Valor no campo	Valor no campo

**Nota:**

- PMLOGO não é válido na chamada MQPUT1 .
- Para o campo *MDMID* , o gerenciador de filas gera um novo identificador de mensagem se PMNMID ou MINONE for especificado e usa o valor no campo, caso contrário.
- Para o campo *MDCID* , o gerenciador de filas gerará um novo identificador de correlação se PMNCID for especificado e usará o valor no campo, caso contrário.

Quando PMLOGO é especificado, o gerenciador de filas requer que todas as mensagens em um grupo e segmentos em uma mensagem lógica sejam colocadas com o mesmo valor no campo



*MDPER* no *MQMD*, ou seja, todas devem ser persistentes ou todas devem ser não persistentes... Se essa condição não for atendida, a chamada *MQPUT* falhará com o código de razão *RC2185*

A opção *PMLOGO* afeta unidades de trabalho da seguinte forma:

- Se a primeira mensagem física em um grupo ou mensagem lógica for colocada em uma unidade de trabalho, todas as outras mensagens físicas no grupo ou mensagem lógica deverão ser colocadas em uma unidade de trabalho, se o mesmo identificador de fila for utilizado. No entanto, eles não precisam ser colocados na mesma unidade de trabalho. Isso permite que um grupo de mensagens ou mensagem lógica que consiste em muitas mensagens físicas seja dividido em duas ou mais unidades consecutivas de trabalho para o identificador de fila.
- Se a primeira mensagem física em um grupo ou mensagem lógica não for colocada em uma unidade de trabalho, nenhuma das outras mensagens físicas no grupo ou mensagem lógica poderá ser colocada em uma unidade de trabalho se o mesmo manipulador de filas for usado.

Se essas condições não forem atendidas, a chamada *MQPUT* falhará com o código de razão *RC2245*

Quando *PMLOGO* é especificado, o *MQMD* fornecido na chamada *MQPUT* não deve ser menor que *MDVER2*. Se essa condição não for atendida, a chamada falhará com o código de razão *RC2257*

Se *PMLOGO* não for especificado, mensagens em grupos e segmentos de mensagens lógicas poderão ser colocadas em qualquer ordem e não será necessário colocar grupos de mensagens completos ou mensagens lógicas completas. É responsabilidade do aplicativo assegurar que os campos *MDGID*, *MDSEQ*, *MDOFF* e *MDMFL* tenham valores apropriados..

Esta é a técnica que pode ser utilizada para reiniciar um grupo de mensagens ou mensagem lógica no meio, após uma falha do sistema ter ocorrido. Quando o sistema for reiniciado, o aplicativo poderá configurar os campos *MDGID*, *MDSEQ*, *MDOFF*, *MDMFL* e *MDPER* para os valores apropriados e, em seguida, emitir a chamada *MQPUT* com *PMSYP* ou *PMNSYP* configurado como *necessário*, mas sem especificar *PMLOGO*. Se essa chamada for bem-sucedida, o gerenciador de filas reterá as informações do grupo e do segmento e as chamadas *MQPUT* subsequentes que usam esse identificador de filas poderão especificar *PMLOGO* como normal

As informações de grupo e segmento que o manipulador de filas retém para a chamada *MQPUT* são separadas das informações de grupo e segmento que ele retém para a chamada *MQGET*.

Para qualquer identificador de fila especificado, o aplicativo é livre para misturar as chamadas *MQPUT* que especificam *PMLOGO* com chamadas *MQPUT* que não o fazem, mas os pontos a seguir devem ser observados:

- Se *PMLOGO* não for especificado, cada chamada *MQPUT* bem-sucedida fará com que o gerenciador de filas configure as informações de grupo e segmento para o identificador de filas para os valores especificados pelo aplicativo; isso substitui as informações de grupo e segmento existentes retidas pelo gerenciador de filas para o identificador de filas.
- Se *PMLOGO* não for especificado, a chamada não falhará se houver um grupo de mensagens ou mensagem lógica atual; no entanto, a chamada poderá ser bem-sucedida com um código de conclusão *CCWARN*. O Tabela 717 na página 1222 mostra os diferentes casos que podem surgir. Nesses casos, se o código de conclusão não for *CCOK*, o código de razão será um dos seguintes (conforme apropriado):
  - *RC2241*
  - *RC2242*
  - *RC2185*
  - *RC2245*

**Nota:** O gerenciador de filas não verifica as informações de grupo e segmento para a chamada *MQPUT1*.

<i>Tabela 717. Resultado quando a chamada MQPUT ou MQCLOSE não é consistente com as informações de grupo e segmento</i>		
<b>A chamada atual é</b>	<b>Chamada anterior era MQPUT com PMLOGO</b>	<b>Chamada anterior era MQPUT sem PMLOGO</b>
MQPUT com PMLOGO	CCFAIL	CCFAIL
MQPUT sem PMLOGO	CCWARN	CCOK
MQCLOSE com um grupo ou mensagem lógica não terminada	CCWARN	CCOK

Os aplicativos que simplesmente desejam colocar mensagens e segmentos em ordem lógica são recomendados para especificar PMLOGO, pois essa é a opção mais simples de usar. Esta opção livra o aplicativo da necessidade de gerenciar as informações de grupo e segmento, pois o gerenciador de filas gerencia essa informação. No entanto, os aplicativos especializados podem precisar de mais controle do que o fornecido pela opção PMLOGO, e isso pode ser alcançado não especificando essa opção. Se isso for feito o aplicativo deve assegurar que os campos *MDGID*, *MDSEQ*, *MDOFF* e *MDMFL* no MQMD sejam configurados corretamente antes de cada chamada MQPUT ou MQPUT1.

Por exemplo, um aplicativo que deseja encaminhar mensagens físicas que recebe, sem considerar se essas mensagens estão em grupos ou segmentos de mensagens lógicas, não deve especificar PMLOGO. Há duas razões para isso:

- Se as mensagens forem recuperadas e colocadas em ordem, especificar PMLOGO faz com que um novo identificador de grupo seja designado às mensagens e isso pode tornar difícil ou impossível para o originador das mensagens correlacionar quaisquer mensagens de resposta ou relatório que resultem do grupo de mensagens.
- Em uma rede complexa com vários caminhos entre os gerenciadores de filas de envio e recebimento, as mensagens físicas podem chegar fora de ordem. Ao não especificar PMLOGO e o GMLOGO correspondente na chamada MQGET, o aplicativo de encaminhamento pode recuperar e encaminhar cada mensagem física assim que ela chegar, sem precisar aguardar a chegada do próximo na ordem lógica...

Aplicativos que geram mensagens de relatório para mensagens em grupos ou segmentos de mensagens lógicas também não devem especificar PMLOGO ao colocar a mensagem de relatório.

PMLOGO pode ser especificado com qualquer uma das outras opções PM\*.

**Opções de contexto:** As opções a seguir controlam o processamento do contexto da mensagem:

#### **PMNOC**

Nenhum contexto deve ser associado à mensagem.

O contexto de identidade e de origem é configurado para não indicar nenhum contexto. Isso significa que os campos de contexto no MQMD são configurados para:

- Espaços em branco para campos de caracteres
- Nulos para campos de bytes
- Zeros para campos numéricos

#### **PMDEFC**

Use o contexto padrão

A mensagem deve ter informações de contexto padrão associadas a ela, para identidade e origem. O gerenciador de filas configura os campos de contexto no descritor de mensagem da seguinte forma:

Tabela 718. Valores de informações de contexto padrão para campos MQMD

<b>Campo no MQMD</b>	<b>Valor Usado</b>
<i>MDUID</i>	Determinado a partir do ambiente, se possível; caso contrário, configure como espaços em branco
<i>MDACC</i>	Determinado a partir do ambiente, se possível; caso contrário, configure como ACNONE
<i>MDAID</i>	Configure como em branco.
<i>MDPAT</i>	Determinado a partir do ambiente.
<i>MDPAN</i>	Determinado a partir do ambiente, se possível; caso contrário, configure como espaços em branco
<i>MDPD</i>	Configure como data quando a mensagem for colocada.
<i>MDPT</i>	Configurar para o horário em que a mensagem é colocada
<i>MDAOD</i>	Configure como em branco.

Para obter mais informações sobre o contexto de mensagem, consulte [Contexto da mensagem](#) e [Informações de contexto de controle](#).

Esta é a ação padrão, se nenhuma opção de contexto for especificada

#### **PMPASI**

Transmitir contexto de identidade a partir de um identificador de filas de entrada.

A mensagem deve ter as informações de contexto associadas a ela O contexto de identidade é obtido do identificador de filas especificado no campo *PMCT* . As informações de contexto de origem são geradas pelo gerenciador de fila da mesma maneira que para *PMDEFEC* (consulte a tabela anterior para obter valores). Para obter mais informações sobre o contexto de mensagem, consulte [Contexto da mensagem](#) e [Informações de contexto de controle](#).

Para a chamada *MQPUT*, a fila deve ter sido aberta com a opção *OOPASI* (ou uma opção que a implica). Para a chamada *MQPUT1* , a mesma verificação de autorização é executada para a chamada *MQOPEN* com a opção *OOPASI*.

#### **PMPASA**

Transmitir todo contexto a partir de um identificador de filas de entrada.

A mensagem deve ter as informações de contexto associadas a ela O contexto de identidade e de origem são obtidos do identificador de filas especificado no campo *PMCT* Para obter mais informações sobre o contexto de mensagem, consulte [Contexto da mensagem](#) e [Informações de contexto de controle](#).

Para a chamada *MQPUT*, a fila deve ter sido aberta com a opção *OOPASA* (ou uma opção que a implica). Para a chamada *MQPUT1* , a mesma verificação de autorização é executada para a chamada *MQOPEN* com a opção *OOPASA*.

#### **PMSETI**

Configurar contexto de identidade a partir do aplicativo.

A mensagem deve ter as informações de contexto associadas a ela O aplicativo especifica o contexto de identidade na estrutura do *MQMD* As informações de contexto de origem são geradas pelo gerenciador de fila da mesma maneira que para *PMDEFEC* (consulte a tabela anterior para obter valores). Para obter mais informações sobre o contexto de mensagem, consulte [Contexto da mensagem](#) e [Informações de contexto de controle](#).

Para a chamada *MQPUT*, a fila deve ter sido aberta com a opção *OOSSETI* (ou uma opção que a implica). Para a chamada *MQPUT1* , a mesma verificação de autorização é executada para a chamada *MQOPEN* com a opção *OOSSETI*.

## PMSETA

Configurar todo contexto a partir do aplicativo.

A mensagem deve ter as informações de contexto associadas a ela. O aplicativo especifica o contexto de identidade e origem na estrutura MQMD. Para obter mais informações sobre o contexto de mensagem, consulte [Contexto da mensagem](#) e [Informações de contexto de controle](#).

Para a chamada MQPUT, a fila deve ter sido aberta com a opção OOSETA. Para a chamada MQPUT1, a mesma verificação de autorização é executada para a chamada MQOPEN com a opção OOSETA.

Apenas uma das opções de contexto PM\* pode ser especificada. Se nenhuma destas opções for especificada, PMDEFC será assumido.

**Tipos de resposta de colocação.** As opções a seguir controlam a resposta retornada para uma chamada MQPUT ou MQPUT1. É possível especificar apenas uma destas opções.. Se PMARES e PMSRES não forem especificados, PMRASQ ou PMRAST será assumido

## PMARES

A opção PMARES solicita que uma operação MQPUT ou MQPUT1 seja concluída sem o aplicativo esperar que o gerenciador de filas conclua a chamada. O uso dessa opção pode melhorar o desempenho do sistema de mensagens, particularmente para aplicativos que usam ligações do cliente. Um aplicativo pode verificar periodicamente, usando o verbo MQSTAT, se ocorreu um erro durante qualquer chamada assíncrona anterior...

Com essa opção, apenas os seguintes campos são garantidos para serem preenchidos no MQMD;

- MDAID
- MDPAT
- MDPAN
- MDAOD

Além disso, se um ou ambos PMNMID ou PMNCID forem especificados como opções, MDMID e MDCID retornados também serão concluídos. (PMNMID pode ser especificado implicitamente, especificando um campo MDMID em branco)

Apenas os campos especificados anteriormente são preenchidos. Outras informações que normalmente seriam retornadas na estrutura MQMD ou MQPMO são indefinidas..

Ao solicitar resposta put assíncrona para MQPUT ou MQPUT1, um CMPCOD e REASON de CCOK e RCNONE não significam necessariamente que a mensagem foi colocada com sucesso em uma fila. Ao desenvolver um aplicativo MQI que usa a resposta put assíncrona e requer a confirmação de que as mensagens foram colocadas em uma fila, você deve verificar os códigos CMPCOD e REASON das operações put e também usar MQSTAT para consultar informações de erro assíncronas

Embora o sucesso ou a falha de cada chamada MQPUT/MQPUT1 individual não possa ser retornado imediatamente, o primeiro erro que ocorreu sob uma chamada assíncrona pode ser determinado em uma junção posterior por meio de uma chamada para MQSTAT.

Se uma mensagem persistente sob o ponto de sincronização falhar ao ser entregue usando a resposta put assíncrona e você tentar confirmar a transação, a confirmação falhará e a transação será restaurada com um código de conclusão de CCFAIL e uma razão de RC2003. O aplicativo pode fazer uma chamada para MQSTAT para determinar a causa de uma falha anterior de MQPUT ou MQPUT1

## PMSRES

Especificar esse valor para uma opção put na estrutura MQPMO assegura que a operação MQPUT ou MQPUT1 seja sempre emitida de forma síncrona. Se a operação for bem-sucedida, todos os campos no MQMD e MQPMO serão concluídos. Ele é fornecido para assegurar uma resposta síncrona independentemente do valor de resposta put padrão definido na fila ou no objeto do tópico.

## **PMRASQ**

Se esse valor for especificado para uma chamada MQPUT, o tipo de resposta put usado será obtido do valor DEFPRESP especificado na fila quando ele foi aberto pelo aplicativo. Se um aplicativo cliente estiver conectado a um gerenciador de fila em um nível anterior ao IBM WebSphere MQ 7.0, ele se comporta como se PMSRES fosse especificado.

Se essa opção for especificada para uma chamada MQPUT1, o valor DEFPRESP da definição de fila não será usado.. Se a chamada MQPUT1 estiver usando PMSYP, ela se comporta como para PMARES e se estiver usando PMNSYP, ela se comporta como para PMSRES.

## **PMRAST**

Este é um sinônimo para PMRASQ para uso com objetos de tópico..

**Outras opções:** As seguintes opções controlam a verificação de autorização e o que acontece quando o gerenciador de filas está em quiesce:

## **PMALTU**

Validar com o identificador de usuário especificado.

Isso indica que o campo *ODAU* no parâmetro **OBJDSC** da chamada MQPUT1 contém um identificador de usuário que deve ser usado para validar a autoridade para colocar mensagens na fila.. A chamada poderá ser bem-sucedida apenas se esse *ODAU* estiver autorizado a abrir a fila com as opções especificadas, independentemente de o identificador de usuário sob o qual o aplicativo está em execução estar autorizado a fazer isso ou não. (Isso não se aplica às opções de contexto especificadas, no entanto, que são sempre verificadas com relação ao identificador de usuário sob o qual o aplicativo está em execução.)..

Esta opção é válida apenas com a chamada MQPUT1 ..

## **PMFIQ**

Falha se o gerenciador de filas estiver quiesce.

Esta opção força a chamada MQPUT ou MQPUT1 a falhar se o gerenciador de filas estiver no estado quiesce.

A chamada retorna o código de conclusão CCFAIL com o código de razão RC2161

**Opção padrão:** se nenhuma das opções descritas anteriormente for necessária, a seguinte opção poderá ser usada:

## **PMNONE**

Nenhuma opção especificada.

Este valor pode ser utilizado para indicar que nenhuma outra opção foi especificada; todas as opções assumem seus valores padrão.. PMNONE é definido para auxiliar a documentação do programa; não se pretende que esta opção seja usada com qualquer outra, mas como seu valor é zero, tal uso não pode ser detectado.

Este é um campo de entrada. O valor inicial do campo *PMOPT* é PMNONE..

## **PMPRF (número inteiro assinado de 10 dígitos)**

Sinalizadores indicando quais campos MQPMR estão presentes.

Este campo contém sinalizadores que devem ser configurados para indicar quais campos MQPMR estão presentes nos registros de mensagens de colocação fornecidos pelo aplicativo *PMPRF* é usado apenas quando a mensagem está sendo colocada em uma lista de distribuição O campo será ignorado se *PMREC* for zero ou ambos *PMPRO* e *PMPRP* forem zero.

Para campos que estão presentes, o gerenciador de filas usa para cada destino os valores dos campos no registro de mensagem put correspondente. Para campos ausentes, o gerenciador de filas usa os valores da estrutura MQMD.

Um ou mais dos sinalizadores a seguir podem ser especificados para indicar quais campos estão presentes nos registros de mensagens put:

## **PFMID**

O campo identificador de mensagem está presente.

**PFCID**

O campo do identificador de correlação está presente.

**PFGID**

O campo identificador de grupo está presente.

**PFFB**

Campo de feedback está presente.

**PFACC**

O campo token de contabilidade está presente.

Se essa sinalização for especificada, PMSETI ou PMSETA deverá ser especificado no campo *PMOPT* ; se essa condição não for satisfeita, a chamada falhará com o código de razão RC2158 .

Se nenhum campo MQPMR estiver presente, o seguinte poderá ser especificado:

**PFNONE**

Nenhum campo de registro de mensagem de colocação está presente

Se esse valor for especificado, *PMREC* deverá ser zero ou *PMPRO* e *PMPRP* deverá ser zero.

PFNONE é definido para a documentação do programa de auxílio Não se pretende que essa constante seja usada com nenhuma outra, mas como seu valor é zero, tal uso não pode ser detectado.

Se *PMPRF* contiver sinalizadores que não são válidos ou registros de mensagens put forem fornecidos, mas *PMPRF* tiver o valor PFNONE, a chamada falhará com o código de razão RC2158 .

Este é um campo de entrada. O valor inicial deste campo é PFNONE. Esse campo será ignorado se *PMVER* for menor que *PMVER2*.

**PMPRO (número inteiro assinado de 10 dígitos)**

Deslocamento do primeiro registro de mensagem put do início de MQPMO.

Este é o deslocamento em bytes do primeiro registro de mensagem MQPMR put a partir do início da estrutura MQPMO. O deslocamento pode ser positivo ou negativo *PMPRO* é usado apenas quando a mensagem está sendo colocada em uma lista de distribuição O campo será ignorado se *PMREC* for zero.

Quando a mensagem está sendo colocada em uma lista de distribuição, uma matriz de um ou mais registros de mensagens de colocação MQPMR pode ser fornecida para especificar determinadas propriedades da mensagem para cada destino individualmente; essas propriedades são:

- identificador de mensagem
- Identificador de correlação
- identificador de grupo
- valor de feedback
- token de conta

Não é necessário especificar todas essas propriedades, mas qualquer subconjunto escolhido, os campos devem ser especificados na ordem correta. Consulte a descrição da estrutura do MQPMR para obter detalhes adicionais

Geralmente, deve haver tantos registros de mensagens put quantos registros de objetos especificados por MQOD quando a lista de distribuição é aberta; cada registro de mensagens put fornece as propriedades de mensagens para a fila identificada pelo registro de objeto correspondente. As filas na lista de distribuição que falham ao abrir ainda devem ter registros de mensagens de colocação alocados para eles nas posições apropriadas na matriz, embora as propriedades de mensagem sejam ignoradas neste caso

É possível que o número de registros de mensagens de colocação seja diferente do número de registros de objetos Se houver menos registros de mensagem put do que registros de objeto, as propriedades de mensagem para os destinos que não possuem registros de mensagem put serão obtidas dos campos correspondentes no descritor de mensagem MQMD Se houver mais registros de

mensagens de colocação do que registros de objetos, o excesso não será usado (embora ainda seja possível acessá-los). Os registros de mensagens put são opcionais, mas se forem fornecidos, deve haver *PMREC* deles.

Os registros de mensagens put podem ser fornecidos de maneira semelhante aos registros de objetos em MQOD, especificando um deslocamento em *PMPRO* ou especificando um endereço em *PMPRP*; para obter detalhes de como fazer isso, consulte o campo *ODORO* descrito em “MQOD (descriptor de objeto) em IBM i” na página 1200..

Não mais de um de *PMPRO* e *PMPRP* pode ser usado; a chamada falhará com o código de razão RC2159 se ambos forem diferentes de zero

Este é um campo de entrada. O valor inicial desse campo é 0. Esse campo será ignorado se *PMVER* for menor que *PMVER2*.

### **PMPRP (ponteiro)**

Endereço do primeiro registro de mensagem put.

Este é o endereço do primeiro registro de mensagem de colocação do MQPMR *PMPRP* é usado apenas quando a mensagem está sendo colocada em uma lista de distribuição. O campo será ignorado se *PMREC* for zero.

*PMPRP* ou *PMPRO* pode ser usado para especificar os registros de mensagens de colocação, mas não ambos; consulte a descrição do campo *PMRRO* para obter detalhes. Se *PMPRP* não for usado, ele deverá ser configurado para o ponteiro nulo ou bytes nulos.

Este é um campo de entrada. O valor inicial deste campo é o indicador nulo. Esse campo será ignorado se *PMVER* for menor que *PMVER2*.

### **PMREC (número inteiro assinado de 10 dígitos)**

Número de registros de mensagens put ou registros de resposta presentes.

Esse é o número de registros de mensagens MQPMR put ou de registros de resposta MQRR que foram fornecidos pelo aplicativo. Esse número pode ser maior que zero somente se a mensagem estiver sendo colocada em uma lista de distribuição.. Registros de mensagens de colocação e registros de resposta são opcionais-o aplicativo não precisa fornecer nenhum registro ou pode optar por fornecer registros de apenas um tipo. Entretanto, se o aplicativo fornecer registros de ambos os tipos, ele deverá fornecer registros *PMREC* de cada tipo.

O valor de *PMREC* não precisa ser igual ao número de destinos na lista de distribuição. Se muitos registros forem fornecidos, o excesso não será usado; se poucos registros forem fornecidos, os valores padrão serão usados para as propriedades de mensagem para os destinos que não tiverem registros de mensagens de colocação (consulte *PMPRO* posteriormente neste tópico).

Se *PMREC* for menor que zero ou maior que zero, mas a mensagem não estiver sendo colocada em uma lista de distribuição, a chamada falhará com o código de razão RC2154 .

Este é um campo de entrada. O valor inicial desse campo é 0. Esse campo será ignorado se *PMVER* for menor que *PMVER2*.

### **PMRMN (sequência de caracteres de 48 bytes)**

Nome resolvido do gerenciador de filas de destino.

Este é o nome do gerenciador de fila de destino após a resolução do nome ter sido executada pelo gerenciador de fila local. O nome retornado é o nome do gerenciador de filas que possui a fila identificada por *PMRQN*. Pode ser o nome do gerenciador de fila local.

Se *PMRQN* for uma fila compartilhada pertencente ao grupo de filas compartilhadas ao qual o gerenciador de filas locais pertence, *PMRMN* será o nome do grupo de filas compartilhadas. Se a fila for de propriedade de algum outro grupo de filas compartilhadas, o *PMRQN* poderá ser o nome do grupo de filas compartilhadas ou o nome de um gerenciador de filas que é um membro do grupo de filas compartilhadas (a natureza do valor retornado é determinada pelas definições de fila que existem no gerenciador de fila local).

Um valor não em branco será retornado apenas se o objeto for uma fila única; se o objeto for uma lista de distribuição ou tópico, o valor retornado será indefinido.

Esse é um campo de saída. O comprimento deste campo é fornecido por LNQM N O valor inicial desse campo é 48 caracteres em branco.

### **PMRQN (sequência de caracteres de 48 bytes)**

Nome da fila de destino resolvido.

Este é o nome da fila de destino após a resolução do nome ter sido executada pelo gerenciador de fila local O nome retornado é o nome de uma fila existente no gerenciador de filas identificado por *PMRMN*.

Um valor não em branco será retornado apenas se o objeto for uma fila única; se o objeto for uma lista de distribuição ou tópico, o valor retornado será indefinido.

Esse é um campo de saída. O comprimento deste campo é fornecido por LNQN (LNQN). O valor inicial desse campo é 48 caracteres em branco.

### **PMRRO (número inteiro assinado de 10 dígitos)**

Deslocamento do primeiro registro de resposta do início de MQPMO.

Este é o deslocamento em bytes do primeiro registro de resposta MQRR do início da estrutura MQPMO. O deslocamento pode ser positivo ou negativo *PMRRO* é usado apenas quando a mensagem está sendo colocada em uma lista de distribuição O campo será ignorado se *PMREC* for zero.

Quando a mensagem está sendo posta em uma lista de distribuição, uma matriz de um ou mais registros de resposta MQRR pode ser fornecida para identificar as filas para as quais a mensagem não foi enviada com sucesso (campo *RRCC* em MQRR) e a razão para cada falha (campo *RRREA* em MQRR). A mensagem pode não ter sido enviada porque a fila falhou ao abrir ou porque a operação put falhou. O gerenciador de filas configura os registros de respostas apenas quando o resultado da chamada é misto (ou seja, algumas mensagens foram enviadas com êxito enquanto outras falharam, ou todas falharam, mas por razões diferentes); o código de razão RC2136 da chamada indica esse caso.. Se o mesmo código de razão se aplicar a todas as filas, esse motivo será retornado no parâmetro **REASON** da chamada MQPUT ou MQPUT1 e os registros de resposta não serão configurados.

Geralmente, deve haver tantos registros de resposta quanto há registros de objeto especificados por MQOD quando a lista de distribuições é aberta; quando necessário, cada registro de resposta é configurado para o código de conclusão e o código de razão para a colocação na fila identificada pelo registro de objeto correspondente As filas na lista de distribuição que falharem ao abrir ainda devem ter registros de resposta alocados para elas nas posições apropriadas na matriz, embora elas estejam configuradas para o código de conclusão e o código de razão resultantes da operação aberta, em vez da operação put.

É possível que o número de registros de resposta seja diferente do número de registros de objetos. Se houver menos registros de resposta do que registros de objeto, pode não ser possível para o aplicativo identificar todos os destinos para os quais a operação put falhou ou as razões para as falhas.. Se houver mais registros de resposta do que registros de objeto, o excesso não será usado (embora ainda seja possível acessá-los). Os registros de resposta são opcionais, mas se forem fornecidos, deverá haver *PMREC* deles.

Os registros de resposta podem ser fornecidos de maneira semelhante aos registros de objeto em MQOD, especificando um deslocamento em *PMRRO* ou especificando um endereço em *PMRRP* ; para obter detalhes de como fazer isso, consulte o campo *ODORO* descrito em “MQOD (descriptor de objeto) em IBM i” na página 1200.. No entanto, não mais de um de *PMRRO* e *PMRRP* pode ser usado; a chamada falha com o código de razão RC2156 se ambos forem diferentes de zero.

Para a chamada MQPUT1 , esse campo deve ser zero. Isso ocorre porque as informações de resposta (se solicitadas) são retornadas nos registros de resposta especificados pelo descriptor de objeto MQOD.

Este é um campo de entrada. O valor inicial desse campo é 0. Esse campo será ignorado se *PMVER* for menor que *PMVER2*.



### **PMRRP (ponteiro)**

Endereço do primeiro registro de resposta..

Este é o endereço do primeiro registro de resposta do MQRR *PMRRP* é usado apenas quando a mensagem está sendo colocada em uma lista de distribuição O campo será ignorado se *PMREC* for zero.

*PMRRP* ou *PMRRO* pode ser usado para especificar os registros de resposta, mas não ambos; consulte a descrição do campo *PMRRO* para obter detalhes.. Se *PMRRP* não for usado, ele deverá ser configurado para o ponteiro nulo ou bytes nulos.

Para a chamada MQPUT1 , esse campo deve ser o ponteiro nulo ou bytes nulos. Isso ocorre porque as informações de resposta (se solicitadas) são retornadas nos registros de resposta especificados pelo descritor de objeto MQOD.

Este é um campo de entrada. O valor inicial deste campo é o indicador nulo. Esse campo será ignorado se *PMVER* for menor que *PMVER2*.

### **PMSID (cadeia de caracteres de 4 bytes)**

Identificador de estruturação.

O valor deve ser:.

#### **PMSIDV**

Identificador para a estrutura de opções put-message.

Isto é sempre um campo de entrada. O valor inicial desse campo é PMSIDV.

### **PMSL (MQLONG)**

O nível de assinatura visado por esta publicação

Apenas as assinaturas com o mais alto *PMSL* menor ou igual a esse valor recebem esta publicação.. Este valor deve estar no intervalo de zero a 9; zero é o nível mais baixo.

O valor inicial deste campo é de 9.

### **PMTO (número inteiro assinado de 10 dígitos)**

Reservado.

Este é um campo reservado; seu valor não é significativo. O valor inicial desse campo é -1.

### **PMUDC (número inteiro assinado de 10 dígitos)**

Número de mensagens enviadas com êxito para as filas remotas

Este é o número de mensagens que a chamada MQPUT ou MQPUT1 atual enviou com êxito para filas na lista de distribuição que são resolvidas para filas remotas. As mensagens que o gerenciador de filas retém temporariamente na lista de distribuição contam como o número de destinos individuais que essas listas de distribuição contêm. Esse campo também é configurado ao colocar uma mensagem em uma única fila que não está na lista de distribuição.

Esse é um campo de saída. O valor inicial desse campo é 0. Esse campo não será configurado se *PMVER* for menor que *PMVER2*.

### **PMVER (número inteiro assinado de 10 dígitos).**

Número de versão da estrutura.

O valor deve ser um dos seguintes:

#### **PMVER1**

Version-1 estrutura de opções put-message.

#### **PMVER2**

Version-2 estrutura de opções put-message.

Os campos que existem apenas na versão mais recente da estrutura são identificados como tal nas descrições dos campos A constante a seguir especifica o número da versão atual:

**PMVERC**

Versão atual da estrutura de opções de put-message

Isto é sempre um campo de entrada. O valor inicial desse campo é PMVER1.

**Valores iniciais**

Tabela 719. Campos em MQPMO		
Nome de Campo	Nome da constante	Valor da constante
PMSID	PMSIDV	'PMO↵'
PMVER	PMVER1	1
PMOPT	PMNONE	0
PMTO	Nenhum	-1
PMCT	Nenhum	0
PMKDC	Nenhum	0
PMUDC	Nenhum	0
PMIDC	Nenhum	0
PMRQN	Nenhum	Espaços em branco
PMRMN	Nenhum	Espaços em branco
PMREC	Nenhum	0
PMPRF	PFNONE	0
PMPRO	Nenhum	0
PMRRO	Nenhum	0
PMPRP	Nenhum	Ponteiro nulo ou bytes nulos
PMRRP	Nenhum	Ponteiro nulo ou bytes nulos
<b>Nota:</b>		
1. O símbolo ↵ representa um único caractere em branco.		

**Declaração RPG**

```

D*..1.....2.....3.....4.....5.....6.....7..
D* MQPMO Structure
D*
D* Structure identifier
D PMSID          1      4    INZ('PMO ')
D* Structure version number
D PMVER          5      8I 0 INZ(1)
D* Options that control the action of MQPUT and MQPUT1
D PMOPT          9      12I 0 INZ(0)
D* Reserved
D PMTO          13     16I 0 INZ(-1)
D* Object handle of input queue
D PMCT          17     20I 0 INZ(0)
D* Number of messages sent successfully to local queues
D PMKDC         21     24I 0 INZ(0)
D* Number of messages sent successfully to remote queues
D PMUDC         25     28I 0 INZ(0)
D* Number of messages that could not be sent

```

```

D PMIDC                29      32I 0 INZ(0)
D* Resolved name of destination queue
D PMRQN                33      80    INZ
D* Resolved name of destination queue manager
D PMRMN                81     128    INZ
D* Number of put message records or response records present
D PMREC                129     132I 0 INZ(0)
D* Flags indicating which MQPMR fields are present
D PMPRF                133     136I 0 INZ(0)
D* Offset of first put message record from start of MQPMO
D PMPRO                137     140I 0 INZ(0)
D* Offset of first response record from start of MQPMO
D PMRRO                141     144I 0 INZ(0)
D* Address of first put message record
D PMPRP                145     160*   INZ(*NULL)
D* Address of first response record
D PMRRP                161     176*   INZ(*NULL)
D* Original message handle
D PMOMH                177     184I 0
D* New message handle
D PMNMH                185     190I 0
D* The action being performed
D PMACT                191     194I 0
D* Reserved
D PMRE1                195     198I 0

```

## IBM i MQPMR (Put-message record) em IBM i

A estrutura MQPMR é usada para especificar várias propriedades de mensagem para um único destino quando uma mensagem está sendo colocada em uma lista de distribuições

### Visão Geral

**Propósito:** MQPMR é uma estrutura de entrada / saída para as chamadas MQPUT e MQPUT1 .

**Conjunto de caracteres e codificação:** os dados em MQPMR devem estar no conjunto de caracteres fornecido pelo atributo do gerenciador de filas **CodedCharSetId** e a codificação do gerenciador de filas locais fornecido por ENNAT. No entanto, se o aplicativo estiver em execução como um cliente IBM MQ , a estrutura deverá estar no conjunto de caracteres e na codificação do cliente..

**Uso:** fornecendo uma matriz dessas estruturas na chamada MQPUT ou MQPUT1 , é possível especificar valores diferentes para cada fila de destino em uma lista de distribuição. Alguns dos campos são somente entrada, outros são entrada/saída.

**Nota:** Essa estrutura é incomum porque não possui um layout fixo. Os campos nesta estrutura são opcionais e a presença ou ausência de cada campo é indicada pelos sinalizadores no campo *PMPRF* em MQPMO Os campos que estão presentes **devem ocorrer na seguinte ordem** :

- *PRMID*
- *PRCID*
- *PRGID*
- *PRFB*
- *PRACC*

Os campos ausentes não ocupam espaço no registro.

Como o MQPMR não possui um layout fixo, nenhuma definição dele é fornecida no arquivo COPY O programador de aplicativos deve criar uma declaração contendo os campos que são requeridos pelo aplicativo e configurar os sinalizadores em *PMPRF* para indicar os campos que estão presentes

- [“Campos” na página 1232](#)
- [“Valores iniciais” na página 1233](#)
- [“Declaração RPG” na página 1233](#)

## Campos

A estrutura MQPMR contém os seguintes campos; os campos são descritos em **ordem alfabética**:

### **PRACC (sequência de bits de 32 bytes).**

Token de Contabilidade.

Este é o token de contabilidade a ser usado para a mensagem enviada para a fila com um nome especificado pelo elemento correspondente na matriz de estruturas MQOR fornecidas na chamada MQOPEN ou MQPUT1 . Ele é processado da mesma forma que o campo *MDACC* no MQMD para uma colocação em uma única fila Consulte a descrição de *MDACC* em “MQMD (Descritor de mensagens) em IBM i” na página 1148 para obter informações sobre o conteúdo desse campo

Se esse campo não estiver presente, o valor em MQMD será usado.

Este é um campo de entrada.

### **PRCID (sequência de bits de 24 bytes)**

Identificador de correlação.

Este é o identificador de correlação a ser usado para a mensagem enviada para a fila com o nome especificado pelo elemento correspondente na matriz de estruturas MQOR fornecidas na chamada MQOPEN ou MQPUT1 . Ele é processado da mesma forma que o campo *MDCID* no MQMD para uma colocação em uma única fila

Se esse campo não estiver presente no registro MQPMR ou houver menos registros MQPMR do que destinos, o valor em MQMD será usado para os destinos que não tiverem um registro MQPMR contendo um campo *PRCID* .

Se *PMNCID* for especificado, um novo identificador de correlação *único* será gerado e usado para todos os destinos na lista de distribuição, independentemente de terem registros MQPMR. Isso é diferente da maneira como *PMNMID* é processado (consulte o campo *PRMID* ).

Este é um campo de entrada/saída

### **PRFB (número inteiro assinado de 10 dígitos)**

Fornecimento de informações ou código de retorno.

Esse é o código de feedback a ser usado para a mensagem enviada para a fila com o nome especificado pelo elemento correspondente na matriz de estruturas MQOR fornecidas na chamada MQOPEN ou MQPUT1 . Ele é processado da mesma forma que o campo *MDFB* no MQMD para uma colocação em uma única fila

Se esse campo não estiver presente, o valor em MQMD será usado.

Este é um campo de entrada.

### **PRGID (sequência de bits de 24 bytes)**

Identificador de grupo.

Este é o identificador de grupo a ser usado para a mensagem enviada para a fila com o nome especificado pelo elemento correspondente na matriz de estruturas MQOR fornecidas na chamada MQOPEN ou MQPUT1 . Ele é processado da mesma forma que o campo *MDGID* no MQMD para uma colocação em uma única fila

Se esse campo não estiver presente no registro MQPMR ou houver menos registros MQPMR do que destinos, o valor em MQMD será usado para os destinos que não tiverem um registro MQPMR contendo um campo *PRGID* . O valor é processado conforme documentado em Tabela 716 na página 1219, mas com as seguintes diferenças:

- Nos casos em que um novo identificador de grupo seria usado, o gerenciador de filas gerará um identificador de grupo diferente para cada destino (ou seja, nenhum dos dois destinos possui o mesmo identificador de grupo).
- Nos casos em que o valor no campo seria usado, a chamada falha com o código de razão RC2258.

Este é um campo de entrada/saída

### PRMID (cadeia de bits de 24 bytes)

Identificador de mensagem.

Este é o identificador de mensagem a ser usado para a mensagem enviada para a fila com o nome especificado pelo elemento correspondente na matriz de estruturas MQOR fornecidas na chamada MQOPEN ou MQPUT1 . Ele é processado da mesma forma que o campo *MDMID* no MQMD para uma colocação em uma única fila

Se esse campo não estiver presente no registro MQPMR ou houver menos registros MQPMR do que destinos, o valor em MQMD será usado para os destinos que não tiverem um registro MQPMR contendo um campo *PRMID* . Se esse valor for MINONE, um novo identificador de mensagem será gerado para *cada* desses destinos (ou seja, nenhum desses destinos tem o mesmo identificador de mensagem).

Se o PMNMID for especificado, novos identificadores de mensagens serão gerados para todos os destinos na lista de distribuição, independentemente de terem registros MQPMR. Isso é diferente da maneira como PMNCID é processado (consulte o campo *PRCID* ).

Este é um campo de entrada/saída

### Valores iniciais

Não há valores iniciais definidos para essa estrutura, pois nenhuma declaração de estrutura é fornecida. A declaração de amostra a seguir mostra como a estrutura deve ser declarada pelo programador do aplicativo, se todos os campos forem necessários

### Declaração RPG

```
D* .1.....2.....3.....4.....5.....6.....7..
D* MQPMR Structure
D*
D* Message identifier
D PRMID 1 24
D* Correlation identifier
D PRCID 25 48
D* Group identifier
D PRGID 49 72
D* Feedback or reason code
D PRFB 73 76I 0
D* Accounting token
D PRACC 77 108
```

## MQRFH (Regras e cabeçalho de formatação) no IBM i

A estrutura MQRFH define o layout das regras e o cabeçalho de formatação

### Visão Geral

**Propósito:** esse cabeçalho pode ser usado para enviar dados de cadeia na forma de pares nome-valor.

**Nome do formato:** FMRFH.

**Conjunto de caracteres e codificação:** os campos na estrutura MQRFH (incluindo *RFNVS*) estão no conjunto de caracteres e codificação fornecidos pelos campos *MDCSI* e *MDENC* na estrutura do cabeçalho que precede o MQRFH ou por esses campos na estrutura MQMD se o MQRFH estiver no início dos dados da mensagem do aplicativo.

O conjunto de caracteres deve ser um que tenha caracteres de byte único para os caracteres válidos em nomes de filas.

- [“Campos” na página 1234](#)
- [“Valores iniciais” na página 1236](#)

- [“Declaração RPG” na página 1236](#)

## Campos

A estrutura MQRFH contém os campos a seguir; os campos são descritos em **ordem alfabética**:

### **RFCSI (número inteiro assinado de 10 dígitos).**

Identificador do conjunto de caracteres de dados que seguem *RFNVS*.

Isso especifica o identificador do conjunto de caracteres dos dados que seguem *RFNVS* ; ele não se aplica aos dados de caracteres na própria estrutura MQRFH

Na chamada MQPUT ou MQPUT1 , o aplicativo deve configurar esse campo para o valor apropriado aos dados. O seguinte valor especial pode ser usado:

#### **CSINHT**

Herde o identificador do conjunto de caracteres desta estrutura..

Os dados de caractere nos dados *após* essa estrutura estão no mesmo conjunto de caracteres que essa estrutura

O gerenciador de filas altera esse valor na estrutura enviada na mensagem para o identificador do conjunto de caractere real da estrutura.. Se nenhum erro ocorrer, o valor CSINHT não será retornado pela chamada MQGET.

CSINHT não poderá ser usado se o valor do campo *MDPAT* em MQMD for ATBRKR.

O valor inicial desse campo é CSUNDF.

Codificação numérica de dados que seguem *RFNVS*..

Isso especifica a codificação numérica dos dados que seguem *RFNVS* ; não se aplica a dados numéricos na própria estrutura MQRFH.

Na chamada MQPUT ou MQPUT1 , o aplicativo deve configurar esse campo para o valor apropriado aos dados.

O valor inicial deste campo é ENNAT

### **RFFLG (número inteiro assinado de 10 dígitos)**

Sinalizadores.

O seguinte pode ser especificado:

#### **RFNONE**

Sem bandeiras.

O valor inicial deste campo é RFNONE..

### **RFFMT (cadeia de caracteres de 8 bytes).**

Formatar o nome dos dados que seguem *RFNVS*

Isso especifica o nome do formato dos dados que seguem *RFNVS*

Na chamada MQPUT ou MQPUT1 , o aplicativo deve configurar esse campo para o valor apropriado aos dados. As regras para codificação desse campo são as mesmas que aquelas para o campo *MDFMT* no MQMD.

O valor inicial deste campo é FMNONE.

### **RFLen (número inteiro assinado de 10 dígitos)**

Comprimento total de MQRFH, incluindo *RFNVS*

Esse é o comprimento em bytes da estrutura MQRFH, incluindo o campo *RFNVS* no final da estrutura. O comprimento não inclui nenhum dado do usuário que siga o campo *RFNVS*

Para evitar problemas com a conversão de dados dos dados do usuário em alguns ambientes, considere usar *RFLEN* como um múltiplo de quatro.

A constante a seguir fornece o comprimento da parte *fixa* da estrutura, ou seja, o comprimento excluindo o campo *RFNVS* :

#### **RFLENV**

Comprimento da parte fixa da estrutura do MQRFH

O valor inicial deste campo é RFLENV.

#### **RFNVS (cadeia de caracteres de n bytes)**

Sequência contendo pares nome-valor.

Esta é uma sequência de caracteres de comprimento variável contendo pares nome-valor no formato:

```
name1 value1 name2 value2 name3 value3 ...
```

Cada nome ou valor deve ser separado do nome ou valor adjacente por um ou mais caracteres em branco; esses espaços em branco não são significativos. Um nome ou valor pode conter espaços em branco significativos prefixando e sufixando o nome ou o valor com o caractere de aspas; todos os caracteres entre aspas de abertura e aspas de fechamento correspondentes são tratados como significativos. No exemplo a seguir, o nome é FAMOUS\_WORDS e o valor é Hello World:

```
FAMOUS_WORDS "Hello World"
```

Um nome ou valor pode conter quaisquer caracteres diferentes do caractere nulo (que age como um delimitador para *RFNVS*). No entanto, para auxiliar na interoperabilidade, um aplicativo pode preferir restringir nomes aos seguintes caracteres:

- Primeiro caractere: alfabético maiúsculo ou minúsculo (A a Z ou a a z) ou sublinhado.
- Caracteres subsequentes: alfabético maiúsculo ou minúsculo, dígito decimal (0 a 9), sublinhado, hífen ou ponto.

Se um nome ou valor contiver uma ou mais aspas, o nome ou valor deverá ser colocado entre aspas e cada aspa dentro da sequência deverá ser dobrada:

```
Famous_Words "The program displayed ""Hello World"""
```

Os nomes e valores fazem distinção entre maiúsculas e minúsculas, ou seja, as letras minúsculas não são consideradas iguais às letras maiúsculas. Por exemplo, FAMOUS\_WORDS e Famous\_Words são dois nomes diferentes.

O comprimento em bytes de *RFNVS* é igual a *RFLEN* menos *RFLENV*. Para evitar problemas com a conversão de dados dos dados do usuário em alguns ambientes, é recomendado que esse comprimento seja um múltiplo de quatro. *RFNVS* deve ser preenchido com espaços em branco para esse comprimento ou finalizado anteriormente colocando um caractere nulo após o último caractere significativo na sequência. O caractere nulo e os bytes após ele, até o comprimento especificado de *RFNVS*, são ignorados.

**Nota:** Como o comprimento desse campo não é fixo, o campo é omitido das declarações da estrutura fornecidas para as linguagens de programação suportadas.

#### **RFSID (sequência de caracteres de 4 bytes).**

Identificador de estruturação.

O valor deve ser:.

#### **RFSIDV**

Identificador para regras e estrutura de cabeçalho de formatação.

O valor inicial desse campo é RFSIDV..

## RFVER (número inteiro assinado de 10 dígitos)

Número de versão da estrutura.

O valor deve ser:.

### RFVER1

Version-1 regras e estrutura do cabeçalho de formatação.

O valor inicial desse campo é RFVER1

## Valores iniciais

Tabela 720. Campos em MQRFH		
Nome de Campo	Nome da constante	Valor da constante
RFSID	RFSIDV	'RFH~'
RFVER	RFVER1	1
RFLEN	RFLNV	32
RFENC	ENNAT	Depende do ambiente
RFCSI	CSUNDF	0
RFFMT	FMNONE	Espaços em branco
RFFLG	RFNONE	0

**Notas:**

1. O símbolo ~ representa um único caractere em branco.

## Declaração RPG

```
D*..1.....2.....3.....4.....5.....6.....7..
D* MQRFH Structure
D*
D* Structure identifier
D RFSID          1      4   INZ('RFH ')
D* Structure version number
D RFVER          5      8I 0 INZ(1)
D* Total length of MQRFH includingNameValueString
D RFLEN          9     12I 0 INZ(32)
D* Numeric encoding of data that followsNameValueString
D RFENC         13     16I 0 INZ(273)
D* Character set identifier of data thatfollows NameValueString
D RFCSI         17     20I 0 INZ(0)
D* Format name of data that followsNameValueString
D RFFMT         21     28   INZ(' ')
D* Flags
D RFFLG         29     32I 0 INZ(0)
```

## MQRFH2 (Regras e cabeçalho de formatação 2) em IBM i

A estrutura MQRFH2 define o formato das regras de version-2 e do cabeçalho de formatação.

### Visão Geral

**Propósito:** Esse cabeçalho pode ser usado para enviar dados que foram codificados usando uma sintaxe semelhante a XML Uma mensagem pode conter duas ou mais estruturas MQRFH2 em série, com dados do usuário opcionalmente seguindo a última estrutura MQRFH2 na série.

**Nome do formato:** FMRFH2.



**Conjunto de caracteres e codificação:** Regras especiais se aplicam ao conjunto de caracteres e codificação usados para a estrutura MQRFH2 :

- Campos diferentes de *RF2NVD* estão no conjunto de caracteres e na codificação fornecidos pelos campos *MDCSI* e *MDENC* na estrutura do cabeçalho que precede MQRFH2 ou por esses campos na estrutura MQMD se o MQRFH2 estiver no início dos dados da mensagem do aplicativo.

O conjunto de caracteres deve ser um que tenha caracteres de byte único para os caracteres válidos em nomes de filas.

Quando GMCONV é especificado na chamada MQGET, o gerenciador de filas converte esses campos para o conjunto de caracteres solicitado e codificação.

- *RF2NVD* está no conjunto de caracteres fornecido pelo campo *RF2NVC* Apenas determinados conjuntos de caracteres Unicode são válidos para *RF2NVC* (consulte a descrição de *RF2NVC* para obter detalhes.

Alguns conjuntos de caracteres têm uma representação que é dependente da codificação. Se *RF2NVC* for um desses conjuntos de caracteres, *RF2NVD* deverá estar na mesma codificação que os outros campos no MQRFH2.

Quando GMCONV for especificado na chamada MQGET, o gerenciador de filas converterá *RF2NVD* para a codificação solicitada, mas não mudará seu conjunto de caracteres

- [“Campos” na página 1237](#)
- [“Valores iniciais” na página 1242](#)
- [“Declaração RPG” na página 1242](#)

## Campos

A estrutura MQRFH2 contém os campos a seguir; os campos são descritos em ordem alfabética:

### **RF2CSI (número inteiro assinado de 10 dígitos)**

O identificador do conjunto de caracteres dos dados que seguem o último campo *RF2NVD* .

Isso especifica o identificador do conjunto de caracteres dos dados que seguem o último campo *RF2NVD* Ele não se aplica aos dados de caractere na própria estrutura MQRFH2

Na chamada MQPUT ou MQPUT1 , o aplicativo deve configurar esse campo para o valor apropriado aos dados. O seguinte valor especial pode ser usado:

#### **CSINHT**

Herde o identificador do conjunto de caracteres desta estrutura..

Os dados de caractere nos dados *após* essa estrutura estão no mesmo conjunto de caracteres que essa estrutura

O gerenciador de filas altera esse valor na estrutura enviada na mensagem para o identificador do conjunto de caractere real da estrutura.. Se nenhum erro ocorrer, o valor CSINHT não será retornado pela chamada MQGET.

CSINHT não poderá ser usado se o valor do campo *MDPAT* em MQMD for ATBRKR.

O valor inicial desse campo é CSINHT.

### **RF2ENC (número inteiro assinado de 10 dígitos)**

Codificação numérica de dados após o último campo *RF2NVD* .

Isso especifica a codificação numérica dos dados que seguem o último campo *RF2NVD* ; ela não se aplica a dados numéricos na própria estrutura MQRFH2

Na chamada MQPUT ou MQPUT1 , o aplicativo deve configurar esse campo para o valor apropriado aos dados.

O valor inicial deste campo é ENNAT

### **RF2FLG (número inteiro assinado de 10 dígitos)**

Sinalizadores.

O seguinte valor deve ser especificado:

#### **RFNONE**

Sem bandeiras.

O valor inicial deste campo é RFNONE..

### **RF2FMT (cadeia de caractere de 8 bytes)**

Nome do formato de dados após o último campo *RF2NVD* .

Isso especifica o nome do formato dos dados que seguem o último campo *RF2NVD*

Na chamada MQPUT ou MQPUT1 , o aplicativo deve configurar esse campo para o valor apropriado aos dados. As regras para codificação desse campo são as mesmas que aquelas para o campo *MDFMT* no MQMD.

O valor inicial deste campo é FMNONE.

### **RF2LEN (número inteiro assinado de 10 dígitos)**

Comprimento total de MQRFH2 incluindo todos os campos *RF2NVL* e *RF2NVD* .

Este é o comprimento em bytes da estrutura MQRFH2 , incluindo os campos *RF2NVL* e *RF2NVD* no final da estrutura. É válido haver vários pares de campos *RF2NVL* e *RF2NVD* no final da estrutura, na sequência:

```
length1, data1, length2, data2, ...
```

O *RF2LEN* não inclui nenhum dado do usuário que possa seguir o último campo *RF2NVD* no final da estrutura

Para evitar problemas com a conversão de dados dos dados do usuário em alguns ambientes, considere usar *RF2LEN* como um múltiplo de quatro.

A constante a seguir fornece o comprimento da parte *fixa* da estrutura, ou seja, o comprimento excluindo os campos *RF2NVL* e *RF2NVD* :

#### **RFLEN2**

Comprimento da parte fixa da estrutura MQRFH2 .

O valor inicial desse campo é RFLEN2

### **RF2NVC (número inteiro assinado de 10 dígitos)**

Identificador do conjunto de caracteres de *RF2NVD*.

Especifica o identificador do conjunto de caracteres codificados dos dados no campo *RF2NVD* . Isso é diferente do conjunto de caracteres das outras sequências na estrutura MQRFH2 e pode ser diferente do conjunto de caracteres dos dados (se houver) que seguem o último campo *RF2NVD* no final da estrutura.

*RF2NVC* deve ter um dos seguintes valores de CCSID:

#### **1200**

UTF-16, a versão Unicode mais recente suportada

#### **13488**

UTF-16, o subconjunto da versão Unicode 2.0

#### **17584**

UTF-16, o subconjunto da versão Unicode 3.0 (inclui o símbolo do euro)

#### **1208**

UTF-8, a versão Unicode mais recente suportada

Para os conjuntos de caracteres UTF-16 , a codificação (ordem de bytes) do *RF2NVD* deve ser igual à codificação dos outros campos na estrutura *MQRFH2* . Caracteres substitutos (X'D800'a X'DFFF') não são suportados.

**Nota:** Se *RF2NVC* não tiver um dos valores listados anteriormente e a estrutura *MQRFH2* requerer conversão na chamada *MQGET*, a chamada será concluída com o código de razão *RC2111* e a mensagem será retornada sem conversão.

O valor inicial deste campo é 1208..

### **RF2NVD (sequência de caracteres de n bytes).**

Dados de nome / valor.

Essa é uma sequência de caracteres de comprimento variável que contém dados codificados usando uma sintaxe semelhante a XML. O comprimento em bytes desta sequência é fornecido pelo campo *RF2NVL* que precede o campo *RF2NVD* ; este comprimento deve ser um múltiplo de quatro.

Os campos *RF2NVL* e *RF2NVD* são opcionais, mas se presentes, eles devem ocorrer como um par e ser adjacentes. O par de campos pode ser repetido quantas vezes forem necessárias, por exemplo:

```
length1 data1 length2 data2 length3 data3
```

Como esses campos são opcionais, eles são omitidos das declarações da estrutura fornecidas para as várias linguagens de programação suportadas.

*RF2NVD* é incomum porque não é convertido no conjunto de caracteres especificado na chamada *MQGET* quando a mensagem é recuperada com a opção *GMCONV* em vigor; *RF2NVD* permanece em seu conjunto de caracteres original. No entanto, *RF2NVD* é convertido na codificação especificada na chamada *MQGET*.

**Sintaxe de dados de nome / valor:** a sequência consiste em uma única "pasta" que contém zero ou mais propriedades. A pasta é delimitada por tags de início e de finalização XML com o mesmo nome da pasta:

```
<folder> property1 property2 ... </folder>
```

Caracteres após a tag de finalização da pasta, até o comprimento definido por *RF2NVL*, devem estar em branco. Na pasta, cada propriedade é composta de um nome e um valor e, opcionalmente, um tipo de dados:

```
<name dt="datatype">value</name>
```

Nestes exemplos:

- Os caracteres delimitadores (<, =, ", / e>) devem ser especificados exatamente conforme mostrado.
- *name* é o nome especificado pelo usuário da propriedade; consulte o exemplo a seguir para obter informações adicionais sobre os nomes.
- *datatype* é um tipo de dados especificado pelo usuário opcional da propriedade; consulte o exemplo a seguir para tipos de dados válidos.
- *value* é o valor especificado pelo usuário da propriedade; consulte os seguintes parágrafos para obter mais informações sobre valores.
- Espaços em branco são significativos entre o caractere > que precede um valor e o caractere < que segue o valor e pelo menos um espaço em branco deve preceder *dt=*. Em outro lugar, os espaços em branco podem ser codificados livremente entre as tags, ou precedendo ou seguindo as tags (por exemplo, para melhorar a legibilidade); esses espaços em branco não são significativos.

Se as propriedades estiverem relacionadas umas às outras, elas poderão ser agrupadas juntas, colocando-as dentro das tags de início e finalização XML com o mesmo nome que o grupo:

```
<folder> <group> property1 property2 ... </group> </folder>
```

Os grupos podem ser aninhados em outros grupos, sem limite, e um grupo pode ocorrer mais de uma vez em uma pasta. Também é válido para uma pasta conter algumas propriedades em grupos e outras propriedades não em grupos.

**Nomes de propriedades, grupos e pastas:** Nomes de propriedades, grupos e pastas devem ser nomes de tags XML válidos, com exceção do caractere dois pontos, que não é permitido em uma propriedade, grupo ou nome da pasta. Principalmente:

- Os nomes devem começar com uma letra ou sublinhado. Letras válidas são definidas na especificação XML W3C e consistem essencialmente em categorias Unicode Ll, Lu, Lo, Lt e Nl.
- Os caracteres restantes em um nome podem ser letras, dígitos decimais, sublinhado, hífen ou pontos. Estes correspondem às categorias Unicode Ll, Lu, Lo, Lt, Nl, Mc, Mn, Lm e Nd.
- Os caracteres de compatibilidade Unicode (X'F900' e acima) não são permitidos em nenhuma parte de um nome..
- Os nomes não devem começar com a cadeia XML em qualquer combinação de letras maiúsculas ou minúsculas

(Co-Marketing Center) ao criar uma MAP. Veja mais detalhes.

- Nomes fazem distinção entre maiúsculas e minúsculas. Por exemplo, ABC, abce Abc são três nomes diferentes.
- Cada pasta possui um namespace separado. Como resultado, um grupo ou propriedade em uma pasta não entra em conflito com um grupo ou propriedade do mesmo nome em outra pasta.
- Grupos e propriedades ocupam o mesmo namespace dentro de uma pasta. Como resultado, uma propriedade não pode ter o mesmo nome que um grupo dentro da pasta que contém essa propriedade

Geralmente, os programas que analisam o campo *RF2NVD* devem ignorar as propriedades ou grupos que possuem nomes que o programa não reconhece, desde que essas propriedades ou grupos sejam formados corretamente

**Tipos de dados de propriedades:** cada propriedade pode ter um tipo de dado opcional... Se especificado, o tipo de dado deve ser um dos seguintes valores, em maiúsculas, minúsculas ou composto por letras maiúsculas e minúsculas:

<b>Tipo de Dados</b>	<b>Usado para</b>
string	Qualquer sequência de caracteres. Determinados caracteres devem ser especificados usando sequências de escape.
boolean	O caractere 0 ou 1 (1 denota TRUE).
bin.hex	Dígitos hexadecimais que representam octetos.
i1	Número inteiro no intervalo de -128 a +127, expresso usando apenas dígitos decimais e sinal opcional..
i2	O número inteiro no intervalo de -32 768 a +32 767, expresso usando apenas dígitos decimais e sinal opcional.
i4	Número inteiro no intervalo de -2 147 483 648 a + 2 147 483 647, expresso usando apenas dígitos decimais e sinal opcional

Tabela 721. Tipos de dados e seu uso (continuação)	
Tipo de Dados	Usado para
i8	Número inteiro no intervalo de -9 223 372 036 854 775 808 a + 9 223 372 036 854 775 807, expresso usando apenas dígitos decimais e sinal opcional..
int	Número inteiro no intervalo de -9 223 372 036 854 775 808 a + 9 223 372 036 854 775 807, expresso usando apenas dígitos decimais e sinal opcional.. Isso pode ser usado no lugar de i1, i2 ou i4 ou i8 se o emissor não desejar sugerir uma precisão específica.
r4	Número de vírgula flutuante com magnitude no intervalo de 1.175E-37 a 3.402 823 47E+38, expresso usando dígitos decimais, sinal opcional, dígitos fracionais opcionais e expoente opcional.
r8	Número de vírgula flutuante com magnitude no intervalo de 2.225E-307 a 1.797 693 134 862 3E+308 expresso usando dígitos decimais, sinal opcional e dígitos fracionários opcionais e expoente opcional.

**Valores de propriedades:** o valor de uma propriedade pode consistir em quaisquer caracteres, exceto para caracteres especiais que possuem uma sequência de escape associada obrigatória. Cada ocorrência no valor de um caractere marcado como "obrigatório" na tabela a seguir deve ser substituída pela sequência de escape correspondente. A tabela também mostra caracteres que possuem uma sequência de escape associada opcional. Cada ocorrência no valor de um caractere marcado como "opcional" pode ser substituída pela sequência de escape correspondente, mas isso não é necessário..

Tabela 722. Caracteres de escape e seu uso		
Caractere	Sequência de escape	Uso
&	&amp;	Obrigatório
<	<	Obrigatório
>	&gt;	Opcional
"	&quot;	Opcional
'	&apos;	Opcional

**Nota:** O caractere & no início de uma sequência de escape não deve ser substituído por &amp; ;

No exemplo a seguir, os espaços em branco no valor são significativos; no entanto, nenhuma sequência de escape é necessária:

```
<Famous_Words>The program displayed "Hello World"</Famous_Words>
```

### RF2NVL (número inteiro assinado de 10 dígitos)

O comprimento de *RF2NVD*

Isso especifica o comprimento em bytes dos dados no campo *RF2NVD*. Para evitar problemas com a conversão de dados (se houver) que segue o campo *RF2NVD*, *RF2NVL* deve ser um múltiplo de quatro.

**Nota:** Os campos *RF2NVL* e *RF2NVD* são opcionais, mas se presentes, eles devem ocorrer como um par e ser adjacentes. O par de campos pode ser repetido quantas vezes forem necessárias, por exemplo:

```
length1 data1 length2 data2 length3 data3
```

Como esses campos são opcionais, eles são omitidos das declarações da estrutura fornecidas para as várias linguagens de programação suportadas.

### **RF2SID (sequência de caracteres de 4 bytes)**

Identificador de estruturação.

O valor deve ser:..

#### **RFSIDV**

Identificador para regras e estrutura de cabeçalho de formatação.

O valor inicial desse campo é RFSIDV..

### **RF2VER (número inteiro assinado de 10 dígitos)**

Número de versão da estrutura.

O valor deve ser:..

#### **RFVER2**

Version-2 regras e estrutura do cabeçalho de formatação

O valor inicial desse campo é RFVER2.

## **Valores iniciais**

<i>Tabela 723. Campos em MQRFH2 .</i>		
<b>Nome de Campo</b>	<b>Nome da constante</b>	<b>Valor da constante</b>
<i>RF2SID</i>	RFSIDV	'RFH↵'
<i>RF2VER</i>	RFVER2	2
<i>RF2LEN</i>	RFLEN2	36
<i>RF2ENC</i>	ENNAT	Depende do ambiente
<i>RF2CSI</i>	CSINHT	-2
<i>RF2FMT</i>	FMNONE	Espaços em branco
<i>RF2FLG</i>	RFNONE	0
<i>RF2NVC</i>	Nenhum	1208

### **Notas:**

1. O símbolo ↵ representa um único caractere em branco.

## **Declaração RPG**

```
D*..1.....2.....3.....4.....5.....6.....7..
D*
D* MQRFH2 Structure
D*
D* Structure identifier
D RF2SID          1      4   INZ('RFH ')
D* Structure version number
D RF2VER          5      8I 0 INZ(2)
D* Total length of MQRFH2 including allNameValueLength and
```

```

D* NameValueDatafields
D RF2LEN          9      12I 0 INZ(36)
D* Numeric encoding of data that followslast NameValueData field
D RF2ENC          13     16I 0 INZ(273)
D* Character set identifier of data thatfollows last NameValueData field
D RF2CSI          17     20I 0 INZ(-2)
D* Format name of data that follows lastNameValueData field
D RF2FMT          21     28     INZ(' ')
D* Flags
D RF2FLG          29     32I 0 INZ(0)
D* Character set identifier ofNameValueData
D RF2NVC          33     36I 0 INZ(1208)

```

IBM i

## MQRMH (Cabeçalho da mensagem de referência) no IBM i

A estrutura MQRMH define o formato de um cabeçalho da mensagem de referência..

### Visão Geral

**Propósito ():** Este cabeçalho é usado com saídas de canal de mensagens gravadas pelo usuário para enviar grandes quantidades de dados (chamados de "dados em massa" ) de um gerenciador de filas para outro. A diferença em relação ao sistema de mensagens normal é que os dados em massa não são armazenados em uma fila; em vez disso, apenas uma *referência* aos dados em massa é armazenada na fila. Isso reduz a possibilidade de recursos do IBM MQ serem esgotados por algumas mensagens grandes.

**Nome do formato:** FMRMH

**Conjunto de caracteres e codificação:** os dados de caracteres em MQRMH e as sequências endereçadas pelos campos de deslocamento devem estar no conjunto de caracteres do gerenciador de fila local; isso é fornecido pelo atributo do gerenciador de filas **CodedCharSetId** .. Dados numéricos em MQRMH devem estar na codificação da máquina nativa; isso é fornecido pelo valor de ENNAT para a linguagem de programação C.

O conjunto de caracteres e a codificação do MQRMH devem ser configurados nos campos *MDCSI* e *MDENC* em:

- O MQMD (se a estrutura MQRMH estiver no início dos dados da mensagem) ou
- A estrutura do cabeçalho que precede a estrutura MQRMH (todos os outros casos).

**Uso:** Um aplicativo coloca uma mensagem consistindo em um MQRMH, mas omitindo os dados em massa.. Quando a mensagem é lida a partir da fila de transmissão por um MCA (Message Channel Agent), uma saída de mensagem fornecida pelo usuário é chamada para processar o cabeçalho da mensagem de referência. A saída pode anexar à mensagem de referência os dados em massa identificados pela estrutura MQRMH, antes que o MCA envie a mensagem através do canal para o próximo gerenciador de filas.

Na extremidade de recebimento, uma saída de mensagem que espera por mensagens de referência deve existir Quando uma mensagem de referência é recebida, a saída deve criar o objeto dos dados em massa que seguem o MQRMH na mensagem e, em seguida, passar a mensagem de referência sem os dados em massa. A mensagem de referência pode ser recuperada posteriormente por um aplicativo que lê a mensagem de referência (sem os dados em massa) a partir de uma fila.

Normalmente, a estrutura MQRMH é tudo o que está na mensagem No entanto, se a mensagem estiver em uma fila de transmissão, um ou mais cabeçalhos adicionais precederão a estrutura MQRMH..

Uma mensagem de referência também pode ser enviada para uma lista de distribuição Neste caso, a estrutura MQDH e seus registros relacionados precedem a estrutura MQRMH quando a mensagem está em uma fila de transmissão.

**Nota:** Uma mensagem de referência não deve ser enviada como uma mensagem segmentada, porque a saída de mensagem não pode processá-la corretamente

- [“Conversão de Dados” na página 1244](#)
- [“Campos” na página 1244](#)

- “Valores iniciais” na página 1248
- “Declaração RPG” na página 1249

## Conversão de Dados

Para propósitos de conversão de dados, a conversão da estrutura MQRMH inclui conversão dos dados do ambiente de origem, nome do objeto de origem, dados do ambiente de destino e nome do objeto de destino. Quaisquer outros bytes dentro de *RMLLEN* bytes do início da estrutura são descartadas ou têm valores indefinidos após a conversão de dados. Os dados em massa serão convertidos desde que todas as instruções a seguir sejam verdadeiras:

- Os dados em massa estão presentes na mensagem quando a conversão de dados é executada..
- O campo *RMFMT* no MQRMH tem um valor diferente de FMNONE.
- Existe uma saída de conversão de dados gravada pelo usuário com o nome do formato especificado....

No entanto, esteja ciente de que geralmente os dados em massa não estão presentes na mensagem quando a mensagem está em uma fila e que, como resultado, os dados em massa não serão convertidos pela opção GMCONV.

## Campos

A estrutura MQRMH contém os campos a seguir; os campos são descritos em **ordem alfabética**:

### RMCSI (número inteiro assinado de 10 dígitos)

Identificador do conjunto de caracteres de dados em massa.

Isso especifica o identificador do conjunto de caracteres dos dados em massa; ele não se aplica aos dados de caracteres na própria estrutura MQRMH

Na chamada MQPUT ou MQPUT1 , o aplicativo deve configurar esse campo para o valor apropriado aos dados. O seguinte valor especial pode ser usado:

#### CSINHT

Herde o identificador do conjunto de caracteres desta estrutura..

Os dados de caractere nos dados *após* essa estrutura estão no mesmo conjunto de caracteres que essa estrutura

O gerenciador de filas altera esse valor na estrutura enviada na mensagem para o identificador do conjunto de caractere real da estrutura.. Se nenhum erro ocorrer, o valor CSINHT não será retornado pela chamada MQGET.

CSINHT não poderá ser usado se o valor do campo *MDPAT* em MQMD for ATBRKR.

O valor inicial desse campo é CSUNDF.

### RMDEL (número inteiro assinado de 10 dígitos)

Comprimento dos dados do ambiente de destino

Se esse campo for zero, não haverá dados do ambiente de destino e *RMDEO* será ignorado.

### RMDEO (número inteiro assinado de 10 dígitos)

Deslocamento de dados do ambiente de destino

Este campo especifica o deslocamento dos dados do ambiente de destino do início da estrutura MQRMH. Os dados do ambiente de destino podem ser especificados pelo criador da mensagem de referência, se esses dados forem conhecidos pelo criador. Por exemplo, os dados do ambiente de destino podem ser o caminho do diretório do objeto no qual os dados em massa devem ser armazenados. No entanto, se o criador não souber os dados do ambiente de destino, será de responsabilidade da saída de mensagem fornecida pelo usuário determinar quaisquer informações de ambiente necessárias



O comprimento dos dados do ambiente de destino é fornecido pelo *RMDEL* ; se esse comprimento for zero, não haverá dados do ambiente de destino e *RMDEO* será ignorado. Se presentes, os dados do ambiente de destino devem residir completamente dentro de *RMLLEN* bytes do início da estrutura.

Os aplicativos não devem assumir que os dados do ambiente de destino são contíguos com qualquer um desses dados endereçados pelos campos *RMSEO*, *RMSNO* e *RMDNO*

O valor inicial desse campo é 0.

#### **RMDL (número inteiro assinado de 10 dígitos)**

Comprimento de dados em massa..

O campo *RMDL* especifica o comprimento dos dados em massa referenciados pela estrutura *MQRMH*.

Se os dados em massa estiverem presentes na mensagem, os dados começarão em um deslocamento de *RMLLEN* bytes do início da estrutura *MQRMH*. O comprimento da mensagem inteira menos *RMLLEN* fornece o comprimento dos dados em massa presentes.

Se os dados estiverem presentes na mensagem, *RMDL* especifica a quantia desses dados que é relevante. O caso normal é que *RMDL* tenha o mesmo valor que o comprimento dos dados presentes na mensagem.

Se a estrutura *MQRMH* representar os dados restantes no objeto (começando a partir do deslocamento lógico especificado), o valor zero poderá ser usado para *RMDL*, se os dados em massa não estiverem presentes na mensagem.

Se nenhum dado estiver presente, o final de *MQRMH* coincide com o final da mensagem.

O valor inicial desse campo é 0.

#### **RMDNL (número inteiro assinado de 10 dígitos)**

O comprimento do nome do objeto de destino

Se esse campo for zero, não haverá nome do objeto de destino e *RMDNO* será ignorado.

#### **RMDNO (número inteiro assinado de 10 dígitos)**

O deslocamento do nome do objeto de destino

Este campo especifica o deslocamento do nome do objeto de destino do início da estrutura *MQRMH*. O nome do objeto de destino pode ser especificado pelo criador da mensagem de referência, se esses dados forem conhecidos pelo criador. No entanto, se o criador não souber o nome do objeto de destino, será responsabilidade da saída de mensagem fornecida pelo usuário identificar o objeto a ser criado ou modificado..

O comprimento do nome do objeto de destino é fornecido por *RMDNL* ; se esse comprimento for zero, não haverá nome de objeto de destino e *RMDNO* será ignorado. Se presente, o nome do objeto de destino deve residir completamente dentro de *RMLLEN* bytes do início da estrutura.

Os aplicativos não devem assumir que o nome do objeto de destino seja contíguo com qualquer um dos dados endereçados pelos campos *RMSEO*, *RMSNO* e *RMDEO*

O valor inicial desse campo é 0.

#### **RMDO (número inteiro assinado de 10 dígitos)**

Deslocamento baixo de dados em massa

Este campo especifica o deslocamento baixo dos dados em massa do início do objeto do qual os dados em massa fazem parte. O deslocamento dos dados em massa do início do objeto é chamado de *deslocamento lógico*. Esse não é o deslocamento físico dos dados em massa do início da estrutura *MQRMH*-esse deslocamento é fornecido por *RMLLEN*.

Para permitir que objetos grandes sejam enviados usando mensagens de referência, o deslocamento lógico é dividido em dois campos e o deslocamento lógico real é fornecido pela soma desses dois campos:

- *RMDO* representa o restante obtido quando o deslocamento lógico é dividido por 1 000 000 000. É, portanto, um valor no intervalo de 0 a 999 999 999.
- *RMDO2* representa o resultado obtido quando o deslocamento lógico é dividido por 1 000 000 000. É, portanto, o número de múltiplos completos de 1 000 000 000 que existem no deslocamento lógico. O número de múltiplos está no intervalo de 0 a 999 999 999.

O valor inicial desse campo é 0.

#### **RMDO2 (número inteiro assinado de 10 dígitos)**

Alto deslocamento de dados em massa

Este campo especifica o alto deslocamento dos dados em massa do início do objeto do qual os dados em massa fazem parte. É um valor no intervalo de 0 a 999 999 999. Consulte *RMDO* para obter detalhes.

O valor inicial desse campo é 0.

#### **RMENC (número inteiro assinado de 10 dígitos)**

Codificação numérica de dados em massa..

Isso especifica a codificação numérica dos dados em massa; ela não se aplica a dados numéricos na própria estrutura MQRMH

Na chamada MQPUT ou MQPUT1 , o aplicativo deve configurar esse campo para o valor apropriado aos dados.

O valor inicial deste campo é ENNAT

#### **RMFLG (número inteiro assinado de 10 dígitos)**

Sinalizadores de mensagem de referência

Os sinalizadores a seguir são definidos:

##### **RMLAST**

A mensagem de referência contém ou representa a última parte do objeto

Esse sinalizador indica que a mensagem de referência representa ou contém a última parte do objeto referido.

##### **RMNLST**

A mensagem de referência não contém ou representa última parte do objeto.

O RMNLST é definido para auxiliar a documentação do programa Não se pretende que essa opção seja usada com qualquer outra, mas como seu valor é zero, tal uso não pode ser detectado.

O valor inicial deste campo é RMNLST.

#### **RMFMT (sequência de caracteres de 8 bytes).**

Nome do formato de dados em massa..

Isso especifica o nome do formato dos dados em massa

Na chamada MQPUT ou MQPUT1 , o aplicativo deve configurar esse campo para o valor apropriado aos dados. As regras para codificação desse campo são as mesmas que aquelas para o campo *MDFMT* no MQMD.

O valor inicial deste campo é FMNONE.

#### **RMLEN (número inteiro assinado de 10 dígitos)**

Comprimento total de MQRMH, incluindo sequências no término de campos fixos, mas não os dados em massa

O valor inicial deste campo é zero.

**RMOII (sequência de bits de 24 bytes).**

Identificador da instância do objeto.

Esse campo pode ser usado na identificação de uma instância específica de um objeto. Se não for necessário, ele deverá ser configurado com o seguinte valor:

**OIINON**

Nenhum identificador de instância do objeto especificado.

O valor é binário zero para o comprimento do campo.

O comprimento deste campo é fornecido por LNOIID. O valor inicial desse campo é OIINON.

**RMOT (cadeia de caracteres de 8 bytes).**

Tipo de objeto.

Este é um nome que pode ser usado pela saída de mensagem para reconhecer os tipos de mensagens de referência que ele suporta. Considere tornar o nome em conformidade com as mesmas regras que o campo *RMFMT*.

O valor inicial deste campo é 8 espaços em branco.

**RMSEL (número inteiro assinado de 10 dígitos)**

Comprimento dos dados do ambiente de origem

Se esse campo for zero, não haverá dados do ambiente de origem e *RMSEO* será ignorado.

O valor inicial desse campo é 0.

**RMSEO (número inteiro assinado de 10 dígitos)**

Compensação de dados do ambiente de origem.

Esse campo especifica o deslocamento dos dados do ambiente de origem do início da estrutura *MQRMH*. Os dados do ambiente de origem podem ser especificados pelo criador da mensagem de referência, se esses dados forem conhecidos pelo criador.. Por exemplo, os dados do ambiente de origem podem ser o caminho de diretório do objeto que contém os dados em massa. No entanto, se o criador não souber os dados do ambiente de origem, será responsabilidade da saída de mensagem fornecida pelo usuário determinar quaisquer informações de ambiente necessárias..

O comprimento dos dados do ambiente de origem é fornecido por *RMSEL*; se esse comprimento for zero, não haverá dados do ambiente de origem e *RMSEO* será ignorado. Se presentes, os dados do ambiente de origem devem residir completamente dentro de *RMLLEN* bytes do início da estrutura.

Os aplicativos não devem assumir que os dados do ambiente são iniciados imediatamente após o último campo fixo da estrutura ou que eles são contíguos com qualquer um dos dados endereçados pelos campos *RMSNO*, *RMDEO* e *RMDNO*.

O valor inicial desse campo é 0.

**RMSID (sequência de caracteres de 4 bytes)**

Identificador de estruturação.

O valor deve ser:.

**RMSIDV**

Identificador para estrutura do cabeçalho da mensagem de referência.

O valor inicial deste campo é RMSIDV.

**RMSNL (número inteiro assinado de 10 dígitos)**

Comprimento do nome do objeto de origem..

Se esse campo for zero, não haverá nome do objeto de origem e *RMSNO* será ignorado.

O valor inicial desse campo é 0.

### **RMSNO (número inteiro assinado de 10 dígitos).**

O deslocamento do nome do objeto de origem

Este campo especifica o deslocamento do nome do objeto de origem do início da estrutura MQRMH. O nome do objeto de origem pode ser especificado pelo criador da mensagem de referência, se esses dados forem conhecidos pelo criador.. No entanto, se o criador não souber o nome do objeto de origem, será de responsabilidade da saída de mensagem fornecida pelo usuário identificar o objeto a ser acessado

O comprimento do nome do objeto de origem é fornecido por *RMSNL* ; se esse comprimento for zero, não haverá nome do objeto de origem e *RMSNO* será ignorado. Se presente, o nome do objeto de origem deve residir completamente dentro de *RMLLEN* bytes do início da estrutura.

Os aplicativos não devem assumir que o nome do objeto de origem seja contíguo com qualquer um dos dados endereçados pelos campos *RMSEO*, *RMDEO* e *RMDNO*

O valor inicial desse campo é 0.

### **RMVER (número inteiro assinado de 10 dígitos)**

Número de versão da estrutura.

O valor deve ser:.

#### **RMVER1**

Version-1 estrutura do cabeçalho da mensagem de referência.

A constante a seguir especifica o número da versão atual:

#### **RMVERC**

A versão atual da estrutura do cabeçalho da mensagem de referência

O valor inicial desse campo é RMVER1.

### **Valores iniciais**

<i>Tabela 724. Campos em MQRMH</i>		
<b>Nome de Campo</b>	<b>Nome da constante</b>	<b>Valor da constante</b>
<i>RMSID</i>	RMSIDV	'RMH↵'
<i>RMVER</i>	RMVER1	1
<i>RMLLEN</i>	Nenhum	0
<i>RMENC</i>	ENNAT	Depende do ambiente
<i>RMCSI</i>	CSUNDF	0
<i>RMFMT</i>	FMNONE	Espaços em branco
<i>RMFLG</i>	RMNLST	0
<i>RMOT</i>	Nenhum	Espaços em branco
<i>RMOII</i>	OIINON	Nulos
<i>RMSEL</i>	Nenhum	0
<i>RMSEO</i>	Nenhum	0
<i>RMSNL</i>	Nenhum	0
<i>RMSNO</i>	Nenhum	0
<i>RMDEL</i>	Nenhum	0
<i>RMDEO</i>	Nenhum	0

Tabela 724. Campos em MQRMH (continuação)

Nome de Campo	Nome da constante	Valor da constante
RMDNL	Nenhum	0
RMDNO	Nenhum	0
RMDL	Nenhum	0
RMDO	Nenhum	0
RMDO2	Nenhum	0

**Notas:**

- O símbolo – representa um único caractere em branco.

```

D*..1.....2.....3.....4.....5.....6.....7..
D*
D* MQRMH Structure
D*
D* Structure identifier
D RMSID          1      4    INZ('RMH ')
D* Structure version number
D RMVER          5      8I 0 INZ(1)
D* Total length of MQRMH, including strings at end of fixed fields, but not
D* the bulk data
D RMLEN          9     12I 0 INZ(0)
D* Numeric encoding of bulk data
D RMENC         13     16I 0 INZ(273)
D* Character set identifier of bulk data
D RMCSI         17     20I 0 INZ(0)
D* Format name of bulk data
D RMFMT         21     28    INZ('      ')
D* Reference message flags
D RMFLG         29     32I 0 INZ(0)
D* Object type
D RMOT          33     40    INZ
D* Object instance identifier
D RMOII         41     64    INZ(X'00000000000000-
D                                     00000000000000000000-
D                                     000000000000')
D* Length of source environment data
D RMSEL         65     68I 0 INZ(0)
D* Offset of source environment data
D RMSEO         69     72I 0 INZ(0)
D* Length of source object name
D RMSNL         73     76I 0 INZ(0)
D* Offset of source object name
D RMSNO         77     80I 0 INZ(0)
D* Length of destination environment data
D RMDL         81     84I 0 INZ(0)
D* Offset of destination environment data
D RMDEO         85     88I 0 INZ(0)
D* Length of destination object name
D RMDNL         89     92I 0 INZ(0)
D* Offset of destination object name
D RMDNO         93     96I 0 INZ(0)
D* Length of bulk data
D RMDL         97     100I 0 INZ(0)
D* Low offset of bulk data
D RMDO         101    104I 0 INZ(0)
D* High offset of bulk data
D RMDO2        105    108I 0 INZ(0)

```

## Declaração RPG

### MQR (Registro de resposta) no IBM i .

A estrutura MQR é usada para receber o código de conclusão e o código de razão resultantes da operação open ou put para uma única fila de destino, quando o destino é uma lista de distribuição.

## Visão Geral

**Propósito:** MQRRE é uma estrutura de saída para chamadas MQOPEN, MQPUT e MQPUT1 .

**Conjunto de caracteres e codificação:** Os dados em MQRRE devem estar no conjunto de caracteres fornecido pelo atributo do gerenciador de fila **CodedCharSetId** e a codificação do gerenciador de filas locais fornecida por ENNAT... No entanto, se o aplicativo estiver em execução como um cliente IBM MQ , a estrutura deverá estar no conjunto de caracteres e na codificação do cliente..

**Uso:** ao fornecer uma matriz dessas estruturas nas chamadas MQOPEN e MQPUT ou na chamada MQPUT1 , é possível determinar os códigos de conclusão e os códigos de razão para todas as filas em uma lista de distribuição quando o resultado da chamada é combinado, ou seja, quando a chamada é bem-sucedida para algumas filas na lista, mas falha para outros.. O código de razão RC2136 da chamada indica que os registros de resposta (se fornecidos pelo aplicativo) foram configurados pelo gerenciador de filas...

- [“Campos” na página 1250](#)
- [“Valores iniciais” na página 1250](#)
- [“Declaração RPG” na página 1250](#)

## Campos

A estrutura MQRRE contém os campos a seguir; os campos são descritos em **ordem alfabética**:

### RRCC (número inteiro assinado de 10 dígitos)

Código de conclusão para fila.

Este é o código de conclusão resultante da operação open ou put para a fila com o nome que foi especificado pelo elemento correspondente na matriz de estruturas MQOR fornecidas na chamada MQOPEN ou MQPUT1 .

Este é sempre um campo de saída O valor inicial desse campo é CCOK.

### RRREA (número inteiro assinado de 10 dígitos)

Código de razão para fila.

Este é o código de razão resultante da operação open ou put para a fila com o nome que foi especificado pelo elemento correspondente na matriz de estruturas MQOR fornecidas na chamada MQOPEN ou MQPUT1 .

Este é sempre um campo de saída O valor inicial deste campo é RCNONE..

## Valores iniciais

Tabela 725. Campos em MQRRE		
Nome de Campo	Nome da constante	Valor da constante
RRCC	CCOK	0
RRREA	RCNONE	0

## Declaração RPG

```
D*..1.....2.....3.....4.....5.....6.....7..
D*
D* MQRRE Structure
D*
D* Completion code for queue
D RRCC          1      4I 0 INZ(0)
D* Reason code for queue
D RRREA        5      8I 0 INZ(0)
```

A estrutura MQSCO (com os campos TLS na estrutura MQCD) permite que um aplicativo em execução como um IBM MQ MQI client especifique opções de configuração que controlam o uso de TLS para a conexão do cliente quando o protocolo do canal é TCP/IP.

## Visão Geral

**Propósito:** A estrutura é um parâmetro de entrada na chamada MQCONNX

Se o protocolo do canal para o canal cliente não for TCP/IP, a estrutura MQSCO será ignorada.

**Conjunto de caracteres e codificação:** Os dados em MQSCO devem estar no conjunto de caracteres fornecido pelo atributo do gerenciador de fila **CodedCharSetId** e a codificação do gerenciador de fila local fornecido por ENNAT.

- [“Campos” na página 1251](#)
- [“Valores iniciais” na página 1255](#)
- [“Declaração RPG” na página 1256](#)

## Campos

A estrutura MQSCO contém os campos a seguir; os campos são descritos em **ordem alfabética**:

### SCAIC (número inteiro assinado de 10 dígitos)

Este é o número de registros de informações sobre autenticação (MQAIR) endereçados pelos campos *SCAIP* ou *SCAIO*. Para obter mais informações, consulte [“MQAIR \(Registro de informações sobre autenticação\) no IBM i” na página 1049](#). O valor deverá ser zero ou maior. Se o valor não for válido, a chamada falha com o código de razão RC2383.

Este é um campo de entrada. O valor inicial desse campo é 0.

### SCAIO (número inteiro assinado de 10 dígitos)

Este é o deslocamento em bytes do primeiro registro de informações sobre autenticação a partir do início da estrutura MQSCO. O deslocamento pode ser positivo ou negativo. O campo será ignorado se *SCAIC* for zero.

É possível usar *SCAIO* ou *SCAIP* para especificar os registros MQAIR, mas não ambos; consulte a descrição do campo *SCAIP* para obter detalhes..

Este é um campo de entrada. O valor inicial desse campo é 0.

### SCAIP (número inteiro assinado de 10 dígitos)

Este é o endereço do primeiro registro de informações sobre autenticação. O campo será ignorado se *SCAIC* for zero.

É possível fornecer a matriz de registros MQAIR de uma de duas maneiras:

- Usando o campo de ponteiro *SCAIP*.

Nesse caso, o aplicativo pode declarar uma matriz de registros MQAIR que é separada da estrutura MQSCO e configurar *SCAIP* para o endereço da matriz.

Considere o uso do *SCAIP* para linguagens de programação que suportam o tipo de dados do ponteiro de uma maneira que seja móvel para ambientes diferentes (por exemplo, a linguagem de programação C).

- Usando o campo de deslocamento *SCAIO*.

Nesse caso, o aplicativo deve declarar uma estrutura composta contendo um MQSCO seguido pela matriz de registros MQAIR e configurar *SCAIO* para o deslocamento do primeiro registro na matriz do início da estrutura MQSCO. Assegure-se de que esse valor esteja correto e tenha um valor que

possa ser acomodado em um MQLONG (a linguagem de programação mais restritiva é COBOL, para o qual o intervalo válido é -999 999 999 a +999 999 999).

Considere o uso do *SCAIO* para linguagens de programação que não suportam o tipo de dados do ponteiro ou que implementam o tipo de dados do ponteiro de forma que não seja móvel para ambientes diferentes (por exemplo, a linguagem de programação COBOL).

Qualquer técnica que você escolher, apenas um de *SCAIP* e *SCAIO* poderá ser usado; a chamada falhará com o código de razão RC2384 se ambos forem diferentes de zero.

Este é um campo de entrada. O valor inicial deste campo é o ponteiro nulo nas linguagens de programação que suportam ponteiros e, caso contrário, uma cadeia de bytes all-null.

**Nota:** Nas plataformas em que a linguagem de programação não suporta o tipo de dados do ponteiro, este campo é declarado como uma cadeia de bytes do comprimento apropriado.

### **SCCERLBL (número inteiro assinado de 10 dígitos)**

Este campo fornece detalhes do rótulo do certificado que está sendo usado

IBM MQ inicializa o valor para o campo SCCERLBL como espaços em branco. Insira o valor necessário ou aceite o valor padrão..

*ibmwebspheremquser\_id* é um valor válido para esse campo para todas as versões do produto e para versões do MQSCO menores que 5.0 é o único valor válido. Portanto, o valor desse campo é interpretado no tempo de execução e alterado, se necessário,. Se você especificar uma versão MQSCO menor que 5.0 ou aceitar o valor padrão de espaços em branco para o campo SCCERLBL, o sistema usará o valor *ibmwebspheremquser\_id*.

Este é um campo de entrada.

### **SCCERTVPOL (número inteiro assinado de 10 dígitos)**

Este campo especifica qual tipo de política de validação de certificado é usado O campo pode ser configurado para um dos seguintes valores:

#### **MQ\_CERT\_VAL\_POLICY\_ANY**

Aplique cada uma das políticas de validação de certificado suportadas pela biblioteca de soquetes seguros.. Aceite a cadeia de certificados se qualquer uma das políticas considerar a cadeia de certificados válida

#### **MQ\_CERT\_VAL\_POLICY\_RFC5280**

Aplique apenas a política de validação de certificado compatível com o RFC5280 Esta configuração fornece validação mais estrita do que a configuração ANY, mas rejeita alguns certificados digitais mais antigos.

O valor inicial desse campo é MQ\_CERT\_VAL\_POLICY\_ANY

### **SCCH (número inteiro assinado de 10 dígitos)**

Este campo fornece detalhes de configuração para o hardware criptográfico conectado ao sistema do cliente

Configure o campo para uma sequência no formato a seguir ou deixe-o em branco ou nulo:

```
GSK_PKCS11=the PKCS #11 driver path and file name;the PKCS #11 token label;the PKCS #11 token password;symmetric cipher setting>;
```

Para usar o hardware criptográfico que está em conformidade com a interface PKCS11 , por exemplo, IBM 4960 ou IBM 4963, especifique o caminho do driver PKCS11 , o rótulo do token PKCS11 e as sequências de senha do token PKCS11 , cada um finalizado por um ponto e vírgula..

O caminho do driver PKCS #11 é um caminho absoluto para a biblioteca compartilhada que fornece suporte para a placa PKCS #11 O nome do arquivo do driver PKCS #11 é o nome da Biblioteca Compartilhada Um exemplo do valor necessário para o nome do caminho e do arquivo PKCS #11 é:



```
/usr/lib/pkcs11/PKCS11_API.so
```

O rótulo do token PKCS #11 deve estar inteiramente em letras minúsculas. Se você tiver configurado seu hardware com um rótulo de token composto por letras maiúsculas ou minúsculas, reconfigure-o com esse rótulo em letras minúsculas.

Se nenhuma configuração de hardware de criptografia for necessária, configure o campo como em branco ou nulo.

Se o valor for menor que o comprimento do campo, termine o valor com um caractere nulo ou preencha-o com espaços em branco até o comprimento do campo. Se o valor não for válido ou levar a uma falha quando usado para configurar o hardware de criptografia, a chamada falhará com o código de razão RC2382.

Este é um campo de entrada. O comprimento deste campo é fornecido por LNSSCH. O valor inicial deste campo é caracteres em branco.

### **CEPSUITEB (número inteiro assinado de 10 dígitos)**

Este campo especifica se a criptografia compatível com o Conjunto B é usada e qual nível de intensidade é empregado. O valor pode ser um ou mais de:

- SCEPSUITEB0  
A criptografia compatível com o Conjunto B não é usada
- SCEPSUITEB1  
A segurança de força do conjunto B de 128 bits é usada
- SCEPSUITEB2  
A segurança de força do conjunto B de 192 bits é usada

**Nota:** O uso de SCEPSUITEB0 com qualquer outro valor nesse campo é inválido.

### **SCFR (número inteiro assinado de 10 dígitos)**

O IBM MQ pode ser configurado com o hardware de criptografia para que os módulos de criptografia usados sejam aqueles fornecidos pelo produto de hardware; eles podem ser certificados pelo FIPS para um nível específico, dependendo do produto de hardware de criptografia em uso.

Use esse campo para especificar que apenas algoritmos certificados por FIPS serão usados, se a criptografia for fornecida no software fornecido pelo IBM MQ.

Quando o IBM MQ é instalado, uma implementação de criptografia TLS também é instalada, o que fornece alguns módulos certificados pelo FIPS.

Os valores podem ser:

#### **MQSSL\_FIPS\_NO**

Esse é o valor-padrão. Quando configurado para este valor:

- Qualquer CipherSpec suportado em uma plataforma específica pode ser usado
- Se for executado sem o uso de hardware de criptografia, o CipherSpec a seguir será executado usando a criptografia certificada FIPS 140-2 nas plataformas IBM MQ :
  - TLS\_RSA\_WITH\_3DES\_EDE\_CBC\_SHA
  - TLS\_RSA\_WITH\_AES\_128\_CBC\_SHA
  - TLS\_RSA\_WITH\_AES\_256\_CBC\_SHA

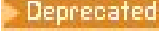
#### **MQSSL\_FIPS\_YES**

Quando configurado para esse valor, a menos que você esteja usando o hardware de criptografia para executar a criptografia, é possível ter certeza de que

- Somente algoritmos criptográficos certificados pelo FIPS podem ser usados no CipherSpec que se aplica a essa conexão do cliente

- As conexões do canal TLS de entrada e de saída serão bem-sucedidas apenas se uma das seguintes Especificações de Cifra for usada:
  - TLS\_RSA\_WITH\_3DES\_EDE\_CBC\_SHA
  - TLS\_RSA\_WITH\_AES\_128\_CBC\_SHA
  - TLS\_RSA\_WITH\_AES\_256\_CBC\_SHA

#### Notas:

1.  CipherSpec TLS\_RSA\_WITH\_3DES\_EDE\_CBC\_SHA foi descontinuado.
2. Quando possível, se CipherSpecs somente FIPS for configurado, o cliente MQI rejeitará conexões que especificam um CipherSpec withRC2393 não FIPS. O IBM MQ não garante rejeitar todas essas conexões e é sua responsabilidade determinar se sua configuração do IBM MQ está com o padrão FIPS.

V 9.3.0

V 9.3.0

#### SCKEYPWL (número inteiro assinado de 10 dígitos)

Este é o comprimento da passphrase do repositório de chaves TLS..

O comprimento máximo da passphrase do repositório de chaves é de 128 caracteres. Se a passphrase do repositório de chaves for maior que o comprimento máximo permitido, a conexão falhará com o RC2381

Este é um campo de entrada. O valor inicial desse campo é 0.

V 9.3.0

V 9.3.0

#### SCKEYPWO (número inteiro assinado de 10 dígitos)

Este é o deslocamento em bytes da passphrase do repositório de chaves TLS O deslocamento pode ser positivo ou negativo

É possível usar SCKEYPWO ou SCKEYPWP para especificar a passphrase do repositório de chave, mas não ambos.. Para obter mais informações, consulte a descrição do campo SCKEYPWP ..

Este é um campo de entrada. O valor inicial desse campo é 0.

V 9.3.0

V 9.3.0

#### SCKEYPWP (ponteiro)

Este é o endereço da passphrase do repositório de chaves TLS..

Este é um campo de entrada. O valor inicial deste campo é o indicador nulo.

A passphrase do repositório de chave pode ser especificada como uma sequência de texto simples ou uma passphrase que foi criptografada usando o utilitário **runmqicred** .

A passphrase do repositório de chave especificada usando esse campo substitui qualquer passphrase do repositório de chave especificada usando a variável de ambiente MQKEYRPWD ou a propriedade SSLKeyRepositoryPassword na sub-rotina SSL do arquivo de configuração do cliente.

É possível usar SCKEYPWO ou SCKEYPWP para especificar a passphrase do repositório de chave, mas não ambos..

#### SCKR (número inteiro assinado de 10 dígitos)

Este campo especifica o local do arquivo do banco de dados de chave no qual as chaves e os

certificados são armazenados   Se o sufixo do arquivo não for especificado, um sufixo `.kdb` será incluído automaticamente.

Cada arquivo do banco de dados de chave pode ter um *arquivo stash de senha* associado Isso mantém senhas criptografadas que são usadas para permitir acesso programático ao banco de dados de chave. O arquivo stash de senha deve residir no mesmo diretório e ter a mesma raiz de arquivo que o banco de dados de chave e deve terminar com o sufixo `.sth`

Por exemplo, se o arquivo do banco de dados de chave for `/xxx/yyy/key.kdb`, o arquivo stash de senha deverá ser `/xxx/yyy/key.sth`, em que `xxx` e `yyy` representarão nomes de diretórios

**V 9.3.0** **V 9.3.0** A senha do banco de dados de chaves também pode ser especificada usando os campos *SCKEYPWP* ou *SCKEYPWO*

Se o valor for menor que o comprimento do campo, termine o valor com um caractere nulo ou preencha-o com espaços em branco até o comprimento do campo. O valor não é verificado; se houver um erro ao acessar o repositório de chaves, a chamada falhará com o código de razão RC2381..

Para executar uma conexão TLS a partir de um IBM MQ MQI client, configure *SCKR* para um nome de arquivo de banco de dados de chaves válido

Este é um campo de entrada. O comprimento deste campo é fornecido por LNSSKR O valor inicial desse campo é um caractere em branco.

### **SCSID (número inteiro assinado de 10 dígitos)**

Este é o identificador de estrutura; o valor deve ser:

#### **SCSIDV**

Identificador para a estrutura de opções de configuração TLS.

Isto é sempre um campo de entrada. O valor inicial desse campo é SCSIDV.

### **SCVER (número inteiro assinado de 10 dígitos)**

Este é o número da versão da estrutura; o valor deve ser:

#### **SCVER1**

Version-1 Estrutura de opções de configuração TLS.

#### **SCVER2**

Estrutura de opções de configuração do TLS Version-2

#### **SCVER3**

Version-3 Estrutura de opções de configuração TLS.

#### **SCVER4**

Version-4 Estrutura de opções de configuração TLS.

#### **SCVER5**

Estrutura de opções de configuração do TLS Version-5

#### **V 9.3.0** **V 9.3.0** **SCVER6**

Estrutura de opções de configuração do TLS Version-6

A constante a seguir especifica o número da versão atual:

#### **SCVERC**







Versão atual da estrutura de opções de configuração do TLS

Isto é sempre um campo de entrada. O valor inicial desse campo é SCVER1.

## **Valores iniciais**

<b>Nome de Campo</b>	<b>Nome da constante</b>	<b>Valor da constante</b>
<i>SCSID</i>	SCSIDV	'SC0~'
<i>SCVER</i>	SCVER1	1
<i>SCKR</i>	Nenhum	Cadeia nula ou espaços em branco
<i>SCCH</i>	Nenhum	Cadeia nula ou espaços em branco
<i>SCAIC</i>	Nenhum	0

Tabela 726. Campos em MQSCO (continuação)

Nome de Campo	Nome da constante	Valor da constante
SCAIO	Nenhum	0
SCAIP	Nenhum	Ponteiro nulo ou bytes nulos
SCKRC	Nenhum	Ponteiro nulo ou bytes nulos
SCFR	Nenhum	Ponteiro nulo ou bytes nulos
SCEPSUITEB	Nenhum	Ponteiro nulo ou bytes nulos
SCCERTVPOL	Nenhum	Ponteiro nulo ou bytes nulos
SCCERLBL	Nenhum	Ponteiro nulo ou bytes nulos
  SCKEYPWP	Nenhum	Ponteiro nulo ou bytes nulos
  SCKEYPWO	Nenhum	0
  SCKEYPWL	Nenhum	0
<b>Notas:</b>		
1. O símbolo ~ representa um único caractere em branco.		
2. Consulte “Declaração RPG” na página 1256 para as opções SCEPSUITEB .		

## Declaração RPG

```

D*.1.....2.....3.....4.....5.....6.....7..
D* MQSCO Structure
D*
D* Structure identifier
D SCSID          1      4    INZ('SCO ')
D* Structure version number
D SCVER          5      8I 0 INZ(1)
D* Location of TLS key repository
D SCKR           9     264  INZ
D* Cryptographic hardware configuration string
D SCCH          265    520  INZ
D* Number of MQAIR records present
D SCAIC         521    524I 0 INZ(0)
D* Offset of first MQAIR record from start of MQSCO structure
D SCAIO         525    528I 0 INZ(0)
D* Address of first MQAIR record
D SCAIP         529    544*  INZ(*NULL)
D* Ver:1 **
D* Number of unencrypted bytes sent/received before secret key is
D* reset
D SCKRC         545    548I 0 INZ(0)
D* Using FIPS-certified algorithms
D SCFR          549    552I 0 INZ(0)
D* Ver:2 **
* Use only Suite B cryptographic algorithms
D SCEPSUITEB0

```

```

D SCEPSUITEB1      553      556I 0 INZ(1)
D SCEPSUITEB2      557      560I 0 INZ(0)
D SCEPSUITEB3      561      564I 0 INZ(0)
D SCEPSUITEB4      565      568I 0 INZ(0)
D SCEPSUITEB       10I 0 DIM(4) OVERLAY(SCEPSUITEB0)
D* Ver:3 **
D* Certificate validation policy
D SCCERTVPOL       569      572I 0 INZ(0)
D* Ver:4 **

```

## IBM i MQSD (Descritores de assinatura) em IBM i

A estrutura MQSD é usada para especificar detalhes sobre a assinatura sendo feita.

### Visão Geral

#### Propósito

A estrutura é um parâmetro de entrada / saída na chamada do MQSUB

#### Assinaturas gerenciadas

Se um aplicativo não tiver necessidade específica de usar uma fila específica como o destino para as publicações que correspondem a sua assinatura, ele poderá usar o recurso de assinatura gerenciada. Se um aplicativo optar por usar uma assinatura gerenciada, o gerenciador de filas informará o assinante sobre o destino para o qual as mensagens publicadas são enviadas, fornecendo uma manipulação de objetos como uma saída da chamada MQSUB. Para obter mais informações, consulte [HOBJ](#) (número inteiro assinado de 10 dígitos)-entrada/saída.

Quando a assinatura é removida, o gerenciador de filas também se compromete a limpar mensagens que não foram recuperadas do destino gerenciado, nas situações a seguir:

- Quando a assinatura é removida-pelo uso de MQCLOSE com CORMSB-e o Hobj gerenciado é fechado.
- Por meios implícitos quando a conexão é perdida para um aplicativo usando uma assinatura não durável (SONDUR)
- Por expiração quando uma assinatura é removida porque expirou e o Hobj gerenciado é fechado.

Você deve usar assinaturas gerenciadas com assinaturas não duráveis, para que a limpeza possa ocorrer e para que as mensagens para assinaturas não duráveis fechadas não ocupem espaço em seu gerenciador de filas. Assinaturas duráveis também podem usar destinos gerenciados.

#### Conjunto de caracteres e codificação

Os dados em MQSD devem estar no conjunto de caracteres fornecido pelo atributo do gerenciador de filas **CodedCharSetId** e pela codificação do gerenciador de fila local fornecido pelo ENNAT. No entanto, se o aplicativo estiver em execução como um cliente IBM MQ, a estrutura deverá estar no conjunto de caracteres e na codificação do cliente..

- [“Campos” na página 1257](#)
- [“Valores iniciais” na página 1270](#)
- [“Declaração RPG” na página 1271](#)

### Campos

A estrutura MQSD contém os campos a seguir; os campos são descritos em ordem alfabética:

#### SDAID (sequência de caracteres de 32 bytes).

Esse valor está no campo *MDAID* do Message Descriptor (MQMD) de todas as mensagens de publicação correspondentes a essa assinatura. *SDAID* faz parte do contexto de identidade da mensagem.. Para obter mais informações sobre contexto da mensagem, consulte [Contexto da mensagem](#).

Para obter mais informações sobre *MDAID*, consulte [MDAID](#).

Se a opção *SOSETI* não for especificada, o *MDAID* que é configurado em cada mensagem publicada para essa assinatura estará em branco, como informações de contextos padrão.

Se a opção *SOSETI* for especificada, o *SDAID* estará sendo gerado pelo usuário e esse campo será um campo de entrada que contém o *MDAID* a ser configurado em cada publicação para essa assinatura

O comprimento deste campo é fornecido por *LNAIDD* O valor inicial desse campo é 32 caracteres em branco.

Se alterar uma assinatura existente usando a opção *SOALT*, o *SDAID* de quaisquer mensagens de publicação futuras poderá ser alterado.

No retorno de uma chamada *MQSUB* usando *SORES*, esse campo é configurado para o atual *MDAID* sendo usado para a assinatura.

### **SDACC (cadeia de caracteres de 32 bytes)**

Esse valor está no campo *MDACC* do Message Descriptor (*MQMD*) de todas as mensagens de publicação correspondentes a essa assinatura. *MDACC* faz parte do contexto de identidade da mensagem.. Para obter mais informações sobre contexto da mensagem, consulte [Contexto da mensagem](#).

Para obter mais informações sobre *MDACC* consulte [MDACC](#).

É possível usar o seguinte valor especial para o campo *SDACC* :

#### **ACNONE**

Nenhum token de contabilidade especificado.

O valor é binário zero para o comprimento do campo.

Se a opção *SOSETI* não for especificada, o token de contabilidade será gerado pelo gerenciador de filas como informações de contextos padrão e esse campo será um campo de saída que contém o *MDACC* que é configurado em cada mensagem publicada para essa assinatura

Se a opção *SOSETI* for especificada, o token de contabilidade está sendo gerado pelo usuário e esse campo é um campo de entrada que contém o *MDACC* a ser configurado em cada publicação para essa assinatura

O comprimento deste campo é fornecido por *LNACCT* O valor inicial desse campo é *ACNONE*.

Se alterar uma assinatura existente usando a opção *SOALT* , o valor de *MDACC* em qualquer mensagem de publicação futura poderá ser alterado.

No retorno de uma chamada *MQSUB* usando *SORES*, esse campo é configurado para o *MDACC* atual que está sendo usado para a assinatura

### **SDASI (cadeia de bits de 40 bytes).**

Este é um identificador de segurança que é transmitido com o *SDAU* para o serviço de autorização para permitir que verificações de autorização apropriadas sejam executadas.

*SDASI* é usado somente se *SOALTU* for especificado e o campo *SDAU* não estiver totalmente em branco até o primeiro caractere nulo ou o final do campo..

No retorno de uma chamada *MQSUB* usando *SORES*, esse campo permanece inalterado.

Consulte a descrição de [ODASI](#) no tipo de dado *MQOD* para obter mais informações

### **SDAU (sequência de caracteres de 12 bytes)**

Se você especificar *SOALTU*, esse campo conterá um identificador de usuário alternativo que será usado para verificar a autorização para a assinatura e a saída para a fila de destino (especificada no parâmetro **Hobj** da chamada *MQSUB*), no lugar do identificador de usuário sob o qual o aplicativo está atualmente em execução.

Se bem-sucedido, o identificador de usuário especificado nesse campo será registrado como o identificador de usuário proprietário da assinatura no lugar do identificador de usuário sob o qual o aplicativo está atualmente em execução.

Se SOALTU for especificado e esse campo estiver inteiramente em branco até o primeiro caractere nulo ou o final do campo, a assinatura poderá ser bem-sucedida somente se nenhuma autorização do usuário for necessária para assinar este tópico com as opções especificadas ou a fila de destino para saída...

Se SOALTU não for especificado, esse campo será ignorado

No retorno de uma chamada MQSUB usando SORES, esse campo permanece inalterado.

Este é um campo de entrada. O comprimento deste campo é fornecido pelo LNUID. O valor inicial desse campo é de 12 caracteres em branco.

### **SDCID (cadeia de bits de 24 bytes)**

Todas as publicações enviadas para corresponder a esta assinatura contêm este identificador de correlação no descritor de mensagem. Se múltiplas assinaturas usarem a mesma fila para obter suas publicações, usar MQGET por ID de correlação permitirá que somente publicações para uma assinatura específica sejam obtidas. Esse identificador de correlação pode ser gerado pelo gerenciador de filas ou pelo usuário.

Se a opção SOSCID não for especificada, o identificador de correlação será gerado pelo gerenciador de filas e esse campo será um campo de saída que contém o identificador de correlação que é configurado em cada mensagem publicada para essa assinatura.

Se a opção SOSCID for especificada, o identificador de correlação será gerado pelo usuário e esse campo será um campo de entrada que contém o identificador de correlação a ser configurado em cada publicação para essa assinatura. Nesse caso, se o campo contiver CINONE, o identificador de correlação que é configurado em cada mensagem publicada para esta assinatura será o identificador de correlação que foi criado pela colocação original da mensagem.

Se a opção SOGRP for especificada e o identificador de correlação especificado for igual a uma assinatura agrupada existente usando a mesma fila e uma sequência de tópico sobreposta, somente a assinatura mais significativa no grupo será fornecida com uma cópia da publicação.

O comprimento desse campo é fornecido por LNCID. O valor inicial desse campo é CINONE.

Se alterar uma assinatura existente usando a opção SOALT e esse campo for um campo de entrada, o ID de correlação da assinatura poderá ser alterado, a menos que a assinatura tenha sido criada usando a opção SOGRP .

No retorno de uma chamada MQSUB usando SORES, esse campo é configurado para o ID de correlação atual para a assinatura.

### **SDEXP (número inteiro assinado de 10 dígitos)**

Este é o tempo expresso em décimos de segundo após o qual a assinatura expira. Nenhuma outra publicação corresponderá a esta subscrição após este intervalo ter passado. Isso também é usado como o valor no campo MDEXP no MQMD das publicações enviadas para este assinante...

O valor especial a seguir é reconhecido:

#### **EIULIM**

A assinatura tem um prazo de expiração ilimitado..

Se alterar uma assinatura existente usando a opção SOALT , a expiração da assinatura poderá ser mudada.

No retorno de uma chamada MQSUB usando a opção SORES , esse campo é configurado para a expiração original da assinatura e não para o tempo de expiração restante

### **SDON (sequência de caracteres de 48 bytes).**

Este é o nome do objeto de tópico conforme definido na gerenciador de filas locais.

O nome pode conter os seguintes caracteres:

- Caracteres alfabéticos maiúsculos (A a Z)
- Caracteres alfabéticos minúsculos (a a z)
- Dígitos numéricos (0 a 9)
- Ponto (.), barra (/), sublinhado (\_), porcentagem (%)

O nome não deve conter espaços em branco iniciais ou integrados, mas pode conter rastreios de espaço em branco. Use um caractere nulo para indicar o término de dados significativos no nome; o nulo e quaisquer caracteres que o seguem são tratados como espaços em branco. As seguintes restrições se aplicam:

- Em sistemas que usam EBCDIC Katakana, caracteres minúsculos não podem ser usados.
- Nomes contendo caracteres minúsculos, barra ou porcentagem devem ser colocados entre aspas quando especificados em comandos. Essas aspas não devem ser especificadas para nomes que ocorrem como campos em estruturas ou como parâmetros em chamadas..

O *SDON* é usado para formar o nome do tópico Completo

O nome completo do tópico pode ser construído de dois campos diferentes: *SDON* e *SDOS*. Para obter detalhes de como esses dois campos são utilizados, consulte [Combinação de seqüências de tópicos](#)

No retorno de uma chamada MQSUB usando a opção SORES , esse campo permanece inalterado.

O comprimento deste campo é fornecido por LNTOPN. O valor inicial desse campo é 48 caracteres em branco.

Se alterar uma assinatura existente usando a opção SDALT, o nome do objeto do tópico inscrito não poderá ser alterado. Esse campo e *SDOS* podem ter sido omitidos Se eles forem fornecidos, eles deverão resolver o mesmo nome do tópico completo ou a chamada falhará com RC2510 .

### **SDOPT (número inteiro assinado de 10 dígitos).**

Você deve especificar pelo menos uma das opções a seguir:

- SOALT
- SORES
- SOCRT

Os valores podem ser incluídos. Não inclua a mesma constante mais de uma vez. A tabela mostra como você pode combinar essas opções: Combinações que não são válidas são anotadas; quaisquer outras combinações são válidas..

#### **Opções de Acesso ou Criação**

As opções de acesso e de criação controlam se uma assinatura é criada ou se uma assinatura existente é retornada ou alterada Deve-se especificar pelo menos uma dessas opções A tabela exibe combinações válidas de opções de acesso ou criação.

<b>Combinação de opções</b>	<b>Notas</b>
SOCRT	Cria uma assinatura se uma não existir; falha se a assinatura existir.
SORES	Continua uma assinatura existente, falha se nenhuma assinatura existir.
SOCRT + SORES	Cria uma assinatura se uma não existir e continua uma correspondente, se existir. Combinação útil, se usada em um aplicativo que pode ser executado várias vezes



<i>Tabela 727. Combinações válidas de opções de acesso e criação (continuação)</i>	
<b>Combinação de opções</b>	<b>Notas</b>
SORES + SOALT (ver nota)	Continua uma assinatura existente, alterando quaisquer campos para corresponder àqueles especificados no MQSD, falhará se não existir nenhuma assinatura
SOCRT + SOALT (ver nota)	Cria uma assinatura se uma não existir e continua uma correspondente, se ela existir, alterando quaisquer campos para corresponder aos especificados no MQSD Combinação útil se usada em um aplicativo que deseja assegurar que sua assinatura esteja em um determinado estado antes de continuar.

**Nota:**

As opções que especificam SOALT também podem especificar SORES, mas essa combinação não tem efeito adicional para especificar SOALT sozinho SOALT implica SORES, pois chamar MQSUB para alterar uma assinatura implica que as assinaturas também são continuadas. O oposto não é verdadeiro, no entanto: retomar uma assinatura não implica que ela deve ser alterada.

**SOCRT**

Crie uma assinatura para o tópico especificado.. Se uma assinatura usando o mesmo *SDSN* existir, a chamada falhará com RC2432 . Essa falha pode ser evitada combinando a opção SOCRT com SORES.. O *SDSN* nem sempre é necessário Para obter mais detalhes, consulte a descrição desse campo.

A combinação de SOCRT com SORES primeiro verifica se há uma assinatura existente para o *SDSN* especificado e se há uma manipulação para essa assinatura preexistente; mas se não houver uma assinatura existente, uma nova será criada usando todos os campos fornecidos no MQSD..

SOCRT também pode ser combinado com SOALT para efeito semelhante (consulte os detalhes sobre SOALT posteriormente neste tópico)..

**SORES**

Retorne um identificador para uma assinatura pré-existente que corresponda àqueles especificadas por *SDSN* Nenhuma mudança é feita para os atributos de assinatura correspondentes e eles são retornados na saída na estrutura MQSD (MQSD). A maior parte do conteúdo do MQSD não é usada: os campos usados são *SDSID*, *SDVER*, *SDOPT*, *SDAID* e *SDASIE* *SDSN*.

A chamada falha com o código de razão RC2428 se uma assinatura não existir correspondente ao nome completo da assinatura. Essa falha pode ser evitada combinando a opção SOCRT com SORES.. Para obter detalhes sobre o SOCRT, consulte [SOCRT](#).

O ID do usuário da assinatura é o ID do usuário que criou a assinatura ou, se ele tiver sido alterado posteriormente por um ID do usuário diferente, ele será o ID do usuário da alteração mais recente e bem-sucedida Se um *SDAID* for usado e o uso de IDs de usuário alternativos for permitido para esse usuário, o *SDAID* será registrado como o ID do usuário que criou a assinatura em vez do ID do usuário sob o qual a assinatura foi feita..

O ID do usuário que criou a assinatura é registrado como *SDAU* se esse campo for usado e o uso de IDs do usuário alternativos for permitido para esse usuário..

Se existir uma assinatura correspondente que tenha sido criada sem a opção SOAUID e o ID do usuário da assinatura for diferente daquele do aplicativo que solicita uma manipulação para a assinatura, a chamada falhará com o código de razão RC2434

Se uma assinatura correspondente existir e estiver atualmente em uso por outro aplicativo, a chamada falhará com o código de razão RC2429 .. Se estiver atualmente em uso pela mesma conexão, a chamada não falhará e um identificador para a assinatura será retornado.

Se a assinatura nomeada em SubName não for uma assinatura válida para continuar ou alterar de um aplicativo, a chamada falhará com RC2523 .

SORES é implícito por SOALT e, portanto, não é necessário ser combinado com essa opção, no entanto, não será um erro se essas duas opções forem combinadas..

## SOALT

Retorne um identificador para uma assinatura preexistente com o nome completo da assinatura correspondente àqueles especificados em SDSN Quaisquer atributos da assinatura que são diferentes daqueles especificados no MQSD são alterados na assinatura, a menos que a alteração seja desaprovada para esse atributo. Os detalhes são anotados na descrição de cada atributo e são resumidos na tabela a seguir: Se você tentar alterar um atributo que não pode ser alterado, a chamada falhará com o código de razão mostrado na tabela a seguir:

A chamada falha com o código de razão RC2428 se uma assinatura não existir correspondente ao nome completo da assinatura. Essa falha pode ser evitada combinando a opção SOCRT com SOALT..

Combinar SOCRT com SOALT primeiro verifica se há uma assinatura existente para o nome da assinatura completa especificado e se há uma manipulação para essa assinatura pré-existente com alterações feitas conforme detalhado anteriormente; mas se não houver nenhuma assinatura existente, uma nova será criada usando todos os campos fornecidos no MQSD...

O ID do usuário da assinatura é o ID do usuário que criou a assinatura ou, se tiver sido alterado posteriormente por um ID do usuário diferente, é o ID do usuário da alteração bem-sucedida mais recente. Se SDAU for usado (e o uso de IDs do usuário alternativos for permitido para esse usuário), o ID do usuário alternativo será registrado como o ID do usuário que criou a assinatura, em vez do ID do usuário sob o qual a assinatura foi feita..

Se existir uma assinatura correspondente que foi criada sem a opção SOAUID e o ID do usuário da assinatura for diferente daquele do aplicativo solicitando uma manipulação para a assinatura, a chamada falhará com o código de razão RC2434 .

Se uma assinatura correspondente existir e estiver atualmente em uso por outro aplicativo, a chamada falhará com RC2429 . Se estiver atualmente em uso pela mesma conexão, a chamada não falhará e um identificador para a assinatura será retornado.

Se a assinatura nomeada em SubName não for uma assinatura válida para continuar ou alterar de um aplicativo, a chamada falhará com RC2523 .

As tabelas a seguir mostram os atributos de subscrição que podem ser alterados pelo SOALT

<i>Tabela 728. Atributos em MQSD e MQSUB que podem ser alterados</i>			
<b>Descritor de tipo de dados ou chamada de função</b>	<b>Nome de Campo</b>	<b>Este atributo pode ser alterado usando SOALT?</b>	<b>Código de razão</b>
MQSD	Opções de durabilidade:	No	RC2509
MQSD	Opções de Destino	Sim	Nenhum
MQSD	Opções de registro	Sim (consulte a nota <a href="#">1</a> )	RC2515 se você tentar alterar SOGRP
MQSD	Opções de publicação	Sim (ver nota <a href="#">2</a> )	Nenhum
MQSD	Opções de curinga	No	RC2510
MQSD	Outras Opções	Não (consulte a nota <a href="#">3</a> )	Nenhum

*Tabela 728. Atributos em MQSD e MQSUB que podem ser alterados (continuação)*

<b>Descritor de tipo de dados ou chamada de função</b>	<b>Nome de Campo</b>	<b>Este atributo pode ser alterado usando SOALT?</b>	<b>Código de razão</b>
MQSD	ObjectName	No	RC2510
MQSD	SDAU	Não (consulte a nota <u>4</u> )	Nenhum
MQSD	SDASI	Não (consulte a nota <u>4</u> )	Nenhum
MQSD	SDEXP	Sim	Nenhum
MQSD	SDOS	No	RC2510
MQSD	SDSN	Não (consulte a nota <u>5</u> )	Nenhum
MQSD	SDSUD	Sim	Nenhum
MQSD	SDCID	Sim (ver nota <u>6</u> )	RC2515 quando em uma assinatura agrupada
MQSD	SDPRI	Sim	Nenhum
MQSD	SDACC	Sim	Nenhum
MQSD	SDAID	Sim	Nenhum
MQSD	SDSL	No	RC2512
MQSUB	Hobj	Sim (ver nota <u>6</u> )	RC2515 quando em uma assinatura agrupada

**Notas:**

1. O SOGRP não pode ser mudado
2. SONEWP não pode ser alterado porque não faz parte da assinatura
3. Essas opções não fazem parte da assinatura
4. Este atributo não faz parte da assinatura
5. Este atributo é a identidade da assinatura sendo alterada
6. Alterável, exceto quando parte de um sub agrupado ( SOGRP )

**Opções de Durabilidade:** As opções a seguir controlam a durabilidade da assinatura. É possível especificar apenas uma destas opções Se estiver alterando uma assinatura existente usando a opção SOALT , não será possível mudar a durabilidade da assinatura. No retorno de uma chamada MQSUB usando SORES, a opção de durabilidade apropriada é configurada..

**SODUR**

Solicite que a assinatura para este tópico permaneça até que ele seja removido explicitamente usando MQCLOSE com a opção CORMSB Se essa assinatura não for removida explicitamente, ela permanecerá mesmo depois que esse aplicativo se conectar ao gerenciador de filas for encerrado

Se uma assinatura durável for solicitada para um tópico definido como não permitindo assinaturas duráveis, a chamada falhará com RC2436 .

**SONDUR**

Solicite que a assinatura deste tópico seja removida quando a conexão do aplicativo com o gerenciador de filas for encerrada, se ainda não tiver sido explicitamente removida. SONDUR é o oposto da opção SODUR e é definido para a documentação do programa de auxílio Ele será o padrão se nenhum for especificado

**Opções de destino:** as opções a seguir controlam o destino para o qual as publicações para um tópico que foi inscrito são enviadas... Se alterar uma assinatura existente usando a opção SOALT, o destino

usado para publicações para a assinatura poderá ser alterado. No retorno de uma chamada MQSUB usando SORES, essa opção é configurada, se apropriado,

## **SOMAN**

Solicite que o destino para o qual as publicações são enviadas seja gerenciado pelo gerenciador de filas

A manipulação de objetos retornada em *HOBj* representa uma fila gerenciada pelo gerenciador de filas e é para uso com as chamadas subseqüentes MQGET, MQCB, MQINQ ou MQCLOSE

Uma manipulação de objetos retornada de uma chamada MQSUB anterior não pode ser fornecida no parâmetro **Hobj** quando o SOMAN não for especificado

**Opções de registro:** As opções a seguir controlam os detalhes do registro que é feito no gerenciador de filas para essa assinatura. Se alterar uma assinatura existente usando a opção SOALT, essas opções de registro poderão ser mudadas. No retorno de uma chamada MQSUB usando SORES, as opções de registro apropriadas são configuradas.

## **SOGRP**

Essa assinatura é agrupada com outras assinaturas do mesmo *SDSL* usando a mesma fila e especificando o mesmo ID de correlação para que quaisquer publicações para tópicos que fariam com que mais de uma mensagem de publicação fosse fornecida para o grupo de assinaturas, devido a um conjunto de sobreposições de seqüências de tópicos que estão sendo usadas, apenas uma mensagem seja entregue para a fila. Se essa opção não for utilizada, cada assinatura exclusiva (identificada por *SDSN*) que corresponde será fornecida com uma cópia da publicação, o que pode significar que mais de uma cópia da publicação pode ser colocada na fila compartilhada por várias assinaturas.

Apenas a assinatura mais significativa do grupo é fornecida com uma cópia da publicação. A assinatura mais significativa é baseada no nome do tópico Completo até o ponto em que um curinga é localizado. Se uma mistura de esquemas curingas for usada dentro do grupo, apenas a posição do curinga será importante. Você é aconselhado a não combinar diferentes esquemas curinga em um grupo de assinaturas que compartilham a mesma fila.

Ao criar uma nova assinatura agrupada, ela ainda deve ter um *SDSN* exclusivo, mas se ela corresponder ao nome completo do tópico de uma assinatura existente no grupo, a chamada falhará com RC2514

Se a assinatura mais significativa no grupo também especificar SONOLC e esta for uma publicação do mesmo aplicativo, nenhuma publicação será entregue para a fila

Ao alterar uma assinatura feita com essa opção, os campos que implicam o agrupamento, *Hobj* na chamada MQSUB (representando a fila e o nome do gerenciador de filas) e o *SDCID* não podem ser alterados. Tentar alterá-los faz com que a chamada falhe com RC2515.

Essa opção deve ser combinada com SOSCID com um *SDCID* que não está configurado como CINONE e não pode ser combinado com SOMAN.

## **SOAUID**

Quando SOAUID for especificado, a identidade do assinante não será restrita a um único ID do usuário. Isso permite que qualquer usuário altere ou continue a assinatura quando tem autoridade adequada. Apenas um único usuário pode ter a assinatura a qualquer momento. Uma tentativa de continuar o uso de uma assinatura atualmente em uso por outro aplicativo faz a chamada falhar com RC2429.

Para incluir essa opção em uma assinatura existente, a chamada MQSUB, usando SOALT, deve vir do mesmo ID do usuário que a própria assinatura original.

Se uma chamada MQSUB fizer referência a uma assinatura existente com SOAUID configurado e o ID do usuário for diferente da assinatura original, a chamada será bem-sucedida somente se o novo ID do usuário tiver autoridade para assinar o tópico. Na conclusão bem-sucedida, as publicações futuras para esse assinante são colocadas na fila do assinante com o novo ID do usuário configurado na mensagem de publicação

Não especifique SOAUID e SOFUID. Se nenhum for especificado, o padrão será SOFUID.

### **SOFUID**

Quando SOFUID é especificado, a assinatura pode ser alterada ou retomada apenas pelo último ID do usuário para alterar a assinatura. Se a assinatura não tiver sido alterada, será o ID do usuário que criou a assinatura.

Se um verbo MQSUB referenciar uma assinatura existente com SOAUID configurado e alterar a assinatura usando SOALT para usar o SOFUID, o ID do usuário da assinatura agora será corrigido nesse novo ID do usuário. A chamada será bem-sucedida apenas se o novo ID do usuário tiver autoridade para assinar o tópico.

Se um ID do usuário diferente daquele registrado como proprietário de uma assinatura tentar continuar ou alterar uma assinatura SOFUID, a chamada falhará com RC2434. O ID do usuário de propriedade de uma assinatura pode ser visualizado usando o comando **DISPLAY SBSTATUS**

Não especifique SOAUID e SOFUID. Se nenhum for especificado, o padrão será SOFUID.

**Opções de publicação:** As opções a seguir controlam a maneira como as publicações são enviadas para este assinante. Se alterar uma assinatura existente usando a opção SOALT, essas opções de publicação poderão ser alteradas.

### **SONOLC**

Informa ao broker que o aplicativo não deseja ver nenhuma de sua própria publicação. As publicações são consideradas originadas do mesmo aplicativo se os identificadores de conexão forem os mesmos. No retorno de uma chamada MQSUB usando SORES, essa opção será configurada, se apropriado,

### **SONEWP**

Nenhuma publicação retida no momento deve ser enviada, quando essa assinatura for criada, apenas novas publicações; Essa opção se aplica somente quando SOCRE é especificado. Quaisquer mudanças subsequentes em uma assinatura não alteram o fluxo de publicações e, portanto, quaisquer publicações que foram retidas em um tópico já foram enviadas ao assinante como novas publicações.

Se essa opção for especificada sem SOCRE, a chamada falhará com RC2046. No retorno de uma chamada MQSUB usando SORES, essa opção não será configurada, mesmo que a assinatura tenha sido criada usando essa opção.

Se essa opção não for usada, as mensagens retidas anteriormente serão enviadas para a fila de destino fornecida. Se essa ação falhar devido a um erro RC2525 ou RC2526, a criação da assinatura falhará.

Essa opção não é válida em combinação com SOPUBR..

### **SOPUBR**

A configuração dessa opção indica que o assinante solicita informações especificamente quando necessário. O gerenciador de filas não envia mensagens não solicitadas ao assinante. A publicação retida (ou possivelmente várias publicações se um curinga for especificado no tópico) é enviada ao assinante cada vez que uma chamada MQSUBRQ é feita usando o identificador Hsub de uma chamada MQSUB anterior. Nenhuma publicação é enviada como um resultado da chamada MQSUB usando esta opção. No retorno de uma chamada MQSUB usando SORES, essa opção será configurada, se apropriado,

Essa opção não é válida em combinação com SONEWP..

**Opções de curinga:** As opções a seguir controlam como os curingas são interpretados na sequência fornecida no campo *SDOS* do MQSD. É possível especificar apenas uma destas opções. Se alterar uma assinatura existente usando a opção SOALT, essas opções de curinga não poderão ser alteradas. No retorno de uma chamada MQSUB usando SORES a opção curinga apropriada é configurada.

### **SOWCHR**

Curingas operam apenas em caracteres dentro da cadeia de tópicos. O campo SOWCHR trata barra (/) como apenas outro caractere sem significado especial.

O comportamento definido por SOWCHR é mostrado na tabela a seguir:

<i>Tabela 729. Como curingas são interpretados</i>	
<b>Caractere especial</b>	<b>Comportamento</b>
*	Curinga, zero ou mais caracteres
?	Curinga, um caractere
%	Caractere de escape para permitir que os caracteres '*', '?' ou '%' sejam usados em uma sequência e não sejam interpretados como um caractere especial, por exemplo, '% *', '%?' ou '%% '.

Por exemplo, publicar no tópico a seguir:

```
/level0/level1/level2/level3/level4
```

corresponde assinantes usando os tópicos a seguir:

```
*
/*
/ level0/level1/level2/level3/*
/ level0/level1/*/level3/level4
/ level0/level1/le?e12/level3/level4
```

**Nota:** Esse uso de curingas fornece exatamente o significado fornecido em IBM MQ V6 e WebSphere MB V6 ao usar mensagens formatadas de MQRFH1 para Publicação / Assinatura. É recomendado que isso não seja usado para aplicativos recém-gravados e seja usado apenas para aplicativos que estavam em execução anteriormente nessa versão e não foram alterados para usar o comportamento curinga padrão, conforme descrito em SOWTOP.

### **SOWTOP**

Os curingas operam apenas em elementos de tópico dentro da sequência de tópicos Este é o comportamento padrão se nenhum for escolhido.

O comportamento requerido pelo SOWTOP é mostrado na tabela a seguir:

<i>Tabela 730. Como curingas são interpretados</i>	
<b>Caractere especial</b>	<b>Comportamento</b>
/	Separador de nível de tópico..
#	Curinga: nível de tópico múltiplo
+	Curinga: nível de tópico único

### **Nota:**

'+' e '#' não serão tratados como curingas se forem combinados com outros caracteres (incluindo eles mesmos) em um nível de tópico. Na sequência a seguir, os caracteres '#' e '+' são tratados como caracteres ordinários.

```
level0/level1/#+/level3/level#
```

Por exemplo, publicar no tópico a seguir:

```
/level0/level1/level2/level3/level4
```

corresponde assinantes usando os tópicos a seguir:

```
#  
/#  
/ level0/level1/level2/level3/#  
/ level0/level1/+level3/level4
```

**Nota:** Esse uso de curingas fornece o significado fornecido no WebSphere Message Broker 6 ao usar mensagens formatadas MQRFH2 para Publicação / Assinatura.

**Outras opções:** as opções a seguir controlam a maneira como a chamada API é emitida em vez da assinatura. No retorno de uma chamada MQSUB usando SORES , essas opções permanecem inalteradas

### SOALTU

O campo SDAU contém um identificador de usuário a ser usado para validar essa chamada MQSUB A chamada pode ser bem-sucedida somente se este SDAU estiver autorizado a abrir o objeto com as opções de acesso especificadas, independentemente de o identificador de usuário sob o qual o aplicativo está em execução estar autorizado a fazer isso..

### SOSCID

A assinatura é para usar o identificador de correlação fornecido no campo *SDCID* Se essa opção não for especificada, um identificador de correlação será criado automaticamente pelo gerenciador de fila no momento da assinatura e será retornado para o aplicativo no campo *SDCID* . Consulte [SDCID \(24-byte bit string\) SDCID](#) para obter mais informações.

### SOSETI

A assinatura é usar o token de contabilidade e dados de identidade do aplicativo fornecidos nos campos *SDACC* e *SDAID* .

Se essa opção for especificada, a mesma verificação de autorização será realizada como se a fila de destino fosse acessada usando uma chamada MQOPEN com 00SETI, exceto no caso em que a opção SOMAN também é usada nesse caso, não há verificação de autorização na fila de destino.

Se essa opção não for especificada, as publicações enviadas para esse assinante possuem informações de contexto padrão associadas a elas da seguinte forma:

Campo no MQMD	Valor Usado
<i>MDUID</i>	O ID do usuário associado à assinatura no momento em que ela foi feita.
<i>MDACC</i>	Determinado a partir do ambiente, se possível; se não, configure como ACNONE.
<i>MDAID</i>	Configurar para espaços em branco..

Essa opção é válida apenas com SOCRE e SOALT. Se usado com SORES, os campos *SDACC* e *SDAID* serão ignorados, portanto, essa opção não tem efeito

Se uma assinatura é alterada sem o uso dessa opção na qual a opção forneceu informações de contexto anteriormente, as informações de contexto padrão são geradas para assinatura alterada.

Se uma assinatura que permite que diferentes IDs de usuário a usem com a opção SOAUIDfor retomada por um ID de usuário diferente, o contexto de identidade padrão será gerado para o novo ID de usuário que agora possui a assinatura e quaisquer publicações subsequentes serão entregues contendo o novo contexto de identidade.

### SOFIQ

A chamada MQSUB falhará se o gerenciador de filas estiver no estado de quiesce No z/OS, para um aplicativo CICS ou IMS , essa opção também força a chamada MQSUB a falhar se a conexão estiver no estado de quiesce

### **SDAU (sequência de caracteres de 12 bytes)**

Se você especificar *SOALTU*, esse campo conterá um identificador de usuário alternativo que será usado para verificar a autorização para a assinatura e a saída para a fila de destino (especificada no parâmetro **Hobj** da chamada *MQSUB*), no lugar do identificador de usuário sob o qual o aplicativo está atualmente em execução.

Se bem-sucedido, o identificador de usuário especificado nesse campo será registrado como o identificador de usuário proprietário da assinatura no lugar do identificador de usuário sob o qual o aplicativo está atualmente em execução.

Se *SOALTU* for especificado e esse campo estiver inteiramente em branco até o primeiro caractere nulo ou o final do campo, a assinatura poderá ser bem-sucedida somente se nenhuma autorização do usuário tiver que assinar este tópico com as opções especificadas ou a fila de destino para saída..

Se *SOALTU* não for especificado, esse campo será ignorado

No retorno de uma chamada *MQSUB* usando *SORES*, esse campo permanece inalterado.

Este é um campo de entrada. O comprimento deste campo é fornecido pelo *LNUID*. O valor inicial desse campo é de 12 caracteres em branco.

### **SDPRI (número inteiro assinado de 10 dígitos).**

Esse é o valor que está no campo *MQPRI* do Message Descriptor (*MQMD*) de todas as mensagens de publicação correspondentes a essa assinatura. Para obter mais informações sobre o campo *MQPRI* no *MQMD*, consulte [MDPR](#).

O valor deve ser maior ou igual a zero; zero é a prioridade mais baixa. Os valores especiais a seguir também podem ser usados:

#### **PRQDEF**

Quando uma fila de assinaturas é fornecida no campo *Hobj* na chamada *MQSUB* e não é um identificador gerenciado, a prioridade para a mensagem é obtida do atributo **DefPriority** dessa fila. Se a fila identificada for uma fila de clusters ou houver mais de uma definição no caminho de resolução do nome da fila, a prioridade será determinada quando a mensagem de publicação for colocada na fila conforme descrito para [MDPRI](#).

Se a chamada *MQSUB* usar uma manipulação gerenciada, a prioridade para a mensagem será obtida do atributo **DefPriority** da fila modelo associada ao tópico inscrito.

#### **PRPUB**

A prioridade para a mensagem é a prioridade da publicação original. Este é o valor inicial do campo

Se alterar uma assinatura existente usando a opção *SOALT*, o *MQPRI* de quaisquer mensagens de publicação futuras poderá ser alterado.

No retorno de uma chamada *MQSUB* usando *SORES*, esse campo é configurado para a prioridade atual sendo usada para a assinatura.

### **SDRO (MQCHARV)**

*SDRO* é o nome do objeto longo depois que o gerenciador de filas resolve o nome fornecido em *SDON*

Se o nome do objeto longo for fornecido em *SDOS* e nada for fornecido em *SDON*, o valor retornado nesse campo será igual ao fornecido em *SDOS*.

Se esse campo for omitido (ou seja, *SDRO.VSBufSize* é zero), o *SDRO* não é retornado, mas o comprimento é retornado no *SDRO.VSLength*. Se o comprimento for menor que o *SDRO* completo, ele será truncado e retornará quantos caracteres mais à direita couber no comprimento fornecido.

Se *SDRO* for especificado incorretamente, de acordo com a descrição de como usar a estrutura [MQCHARV](#) ou se exceder o comprimento máximo, a chamada falhará com o código de razão RC2520.

### **SDSID (sequência de caracteres de 4 bytes).**

Este é o identificador de estrutura; o valor deve ser:



## **SDSIDV**

Identificador para a estrutura do Descritor de Assinatura.

Isto é sempre um campo de entrada. O valor inicial desse campo é SDSIDV

## **SDSL (número inteiro assinado de 10 dígitos)**

Esse é o nível associado à assinatura. As publicações serão entregues para esta assinatura somente se estiverem no conjunto de assinaturas com o valor *SDSL* mais alto menor ou igual ao *PubLevel* usado no momento da publicação.

O valor deve estar no intervalo de zero a 9 Zero é o nível mais baixo.

O valor inicial desse campo é 1.

Se alterar uma assinatura existente usando a opção *SOALT* , *SDSL* não poderá ser mudado.

## **SDSN (MÇCHARV)**

*SDSN* especifica o nome da subscrição

Esse campo será necessário apenas se *SDOPT* especificar a opção *SODUR* , mas se for fornecido, ele também será usado pelo gerenciador de filas para *SONDUR* Se especificado, *SDSN* deve ser exclusivo no gerenciador de filas, porque é o campo usado para identificar assinaturas.

O comprimento máximo de *SDSN* é 10240.

Este campo tem dois propósitos. Para uma assinatura *SODUR*, é o meio pelo qual você identifica uma assinatura para retomá-la após ela ter sido criada, se você tiver fechado o identificador para a assinatura (usando a opção *COKPSB* ) ou tiver sido desconectado do gerenciador de filas. A identificação de uma assinatura para removê-la após sua criação é feita usando a chamada *MQSUB* com a opção *SORES* . O campo *SDSN* também é exibido na visualização de administração de assinaturas no campo *SDSN* em *DISPLAY SBSTATUS*

Se *SDSN* for especificado incorretamente, de acordo com a descrição de como usar a estrutura *MÇCHARV* ou se exceder o comprimento máximo, ou se for omitido quando for necessário (ou seja, *SDSN.VCHRL* é zero) ou, se exceder o comprimento máximo, a chamada falhará com o código de razão *RC2440* .

Este é um campo de entrada. Os valores iniciais dos campos nessa estrutura são os mesmos da estrutura *MÇCHARV*.

Se alterar uma assinatura existente usando a opção *SOALT* , o nome da assinatura não poderá ser alterado, porque ele é o campo usado para identificar a assinatura. Ela não é mudada na saída de uma chamada *MQSUB* com a opção *SORES*

## **SDSS (MÇCHARV)**

*SDSS* é a sequência que fornece os critérios de seleção usados ao assinar mensagens de um tópicos.

Esse campo de comprimento variável será retornado na saída de uma chamada *MQSUB* usando a opção *SORES* , se um buffer for fornecido e se também houver um comprimento de buffer positivo em *VSBufSize*. Se nenhum buffer for fornecido na chamada, apenas o comprimento da sequência de seleção será retornado no campo *VSLength* do *MÇCHARV* Se o buffer fornecido for menor que o espaço necessário para retornar o campo, somente *VSBufSize* bytes serão retornados no buffer fornecido.

Se *SDSS* for especificado incorretamente, de acordo com a descrição de como usar a estrutura *MÇCHARV* ou se exceder o comprimento máximo, a chamada falhará com o código de razão *RC2519* .

## **SDSUD (MÇCHARV)**

Os dados fornecidos sobre a assinatura neste campo são incluídos como a propriedade de mensagem *mq.SubUserData* de cada publicação enviada para esta assinatura

O comprimento máximo de *SDSUD* é 10240.

Se *SDSUD* for especificado incorretamente, de acordo com a descrição de como usar a estrutura *MÇCHARV* ou se exceder o comprimento máximo, a chamada falhará com o código de razão *RC2431*.

Este é um campo de entrada. Os valores iniciais dos campos nessa estrutura são os mesmos da estrutura MQCHARV.

Se alterar uma assinatura existente usando a opção SOALT , os dados do usuário da assinatura poderão ser alterados.

Esse campo de comprimento variável é retornado na saída de uma chamadas MQSUB usando a opção SORES , se um buffer for fornecido e houver um comprimento de buffer positivo em *VSBufLen*. Se nenhum buffer for fornecido na chamada, apenas o comprimento dos dados do usuário da assinatura será retornado no campo *VCHRL* de MQCHARV Se o buffer fornecido for menor que o espaço necessário para retornar o campo, apenas *VSBufLen* bytes serão retornados no buffer fornecido..

### **SDVER (número inteiro assinado de 10 dígitos).**

Este é o número da versão da estrutura; o valor deve ser:

#### **SDVER1**

Version-1 Estrutura do descritor de assinatura.

A constante a seguir especifica o número da versão atual:

#### **SDVERC**

Versão atual da estrutura do Descritor de Subscrição

Isto é sempre um campo de entrada. O valor inicial do campo é SDVER1.

### **Valores iniciais**

<i>Tabela 732. Campos em MQSD</i>		
<b>Nome de Campo</b>	<b>Nome da constante</b>	<b>Valor da constante</b>
<i>SDSID</i>	SDSIDV	'SD--'
<i>SDVER</i>	SDVER1	1
<i>SDOPT</i>	SONDUR	0
<i>SDON</i>	Nenhum	Espaços em branco
<i>SDAU</i>	Nenhum	Espaços em branco
<i>SDASI</i>	SINONE	Nulos
<i>SDEXP</i>	EIULIM	-1
<i>SDOS</i>	Nomes e valores conforme definidos para MQCHARV	
<i>SDSN</i>	Nomes e valores conforme definidos para MQCHARV	
<i>SDSUD</i>	Nomes e valores conforme definidos para MQCHARV	
<i>SDCID</i>	CINONE	Nulos
<i>SDPRI</i>	PRQDEF	-3
<i>SDACC</i>	ACNONE	Nulos
<i>SDAID</i>	Nenhum	Espaços em branco
<i>SDSL</i>	Nenhum	1
<i>SDRO</i>	Nomes e valores conforme definidos em MQCHARV	

Tabela 732. Campos em MQSD (continuação)

Nome de Campo	Nome da constante	Valor da constante
<b>Nota:</b>		
1. O símbolo ~ representa um único caractere em branco.		

## Declaração RPG

```

D*..1.....2.....3.....4.....5.....6.....7..
D* MQSD Structure
D*
D* Structure identifier
D SDSID          1          4
D* Structure version number
D SDVER          5          8I 0
D* Options associated with subscribing
D SDOPT          9          12I 0
D* Object name
D SDON           13         60
D* Alternate user identifier
D SDAU           61         72
D* Alternate security identifier
D SDASI          73         112
D* Expiry of Subscription
D SDEXP          113        116I 0
D* Object Long name
D SDOSP          117        132*
D SDOSO          133        136I 0
D SDOSS          137        140I 0
D SDOSL          141        144I 0
D SDOSC          145        148I 0
D* Subscription name
D SDSNP          149        164*
D SDSNO          165        168I 0
D SDSNS          169        172I 0
D SDSNL          173        176I 0
D SDSNC          177        180I 0
D* Subscription User data
D SDSUDP         181        196*
D SDSUDO         197        200I 0
D SDSUDS         201        204I 0
D SDSUDL         205        208I 0
D SDSUDC         209        212I 0
D* Correlation Id related to this subscription
D SDCID          213        236
D* Priority set in publications
D SDPRI          237        240I 0
D* Accounting Token set in publications
D SDACC          241        272
D* Appl Identity Data set in publications
D SDAID          273        304
D* Message Selector
D SDSSP          305        320*
D SDSSO          321        324I 0
D SDSSS          325        328I 0
D SDSSL          329        332I 0
D SDSSC          333        336
D* Subscription level
D SDSL           337        340 0
D* Resolved Long object name
D SDROP          341        356*
D SDR00          357        360I 0
D SDR0S          361        364I 0
D SDR0L          365        368I 0
D SDR0C          369        372I 0

```

## IBM i MQSMPO (Configurar opções de propriedade de mensagem) em IBM i.

A estrutura **MQSMPO** permite que aplicativos especifiquem opções que controlam como as propriedades de mensagens são configuradas.

## Visão Geral

**Propósito:** A estrutura é um parâmetro de entrada na chamada **MQSETMP**

**Conjunto de caracteres e codificação:** os dados em **MQSMPO** devem estar no conjunto de caracteres do aplicativo e na codificação do aplicativo (ENNAT).

- “Campos” na página 1272
- “Valores iniciais” na página 1273
- “Declaração RPG” na página 1273

## Campos

A estrutura MQSMPO contém os campos a seguir; os campos são descritos em **ordem alfabética**:

### **SPOPT (número inteiro assinado de 10 dígitos)**

**Opções de Local:** As seguintes opções estão relacionadas ao local relativo da propriedade em comparação com o cursor da propriedade:

#### **SPSETF**

Configura o valor da primeira propriedade que corresponde ao nome especificado ou, se não existir, inclui uma nova propriedade após todas as outras propriedades com uma hierarquia correspondente.

#### **SPSETC**

Configura o valor da propriedade apontada pelo cursor de propriedade A propriedade apontada pelo cursor da propriedade é aquela que foi consultada pela última vez usando a opção IPINQF ou IPINQN..

O cursor de propriedade é reconfigurado quando a manipulação de mensagem é reutilizada, ou quando a manipulação de mensagem é especificada no campo *HMSG* da estrutura MQGMO em uma chamada MQGET ou a estrutura MQPMO em uma chamada MQPUT

Se essa opção for usada quando o cursor de propriedade ainda não tiver sido estabelecido, ou se a propriedade apontada pelo cursor de propriedade tiver sido excluída, a chamada falhará com o código de conclusão CCFAIL e o código de razão RC2471

#### **SPSETA**

Configura uma nova propriedade após a propriedade apontada pelo cursor da propriedade A propriedade apontada pelo cursor de propriedade é aquela que foi consultada pela última vez usando a opção IPINQF ou IPINQO

O cursor de propriedade é reconfigurado quando a manipulação de mensagem é reutilizada, ou quando a manipulação de mensagem é especificada no campo *HMSG* da estrutura MQGMO em uma chamada MQGET ou a estrutura MQPMO em uma chamada MQPUT

Se essa opção for usada quando o cursor de propriedade ainda não tiver sido estabelecido, ou se a propriedade apontada pelo cursor de propriedade tiver sido excluída, a chamada falhará com o código de conclusão CCFAIL e o código de razão RC2471

Se você não precisar de nenhuma das opções descritas, use a seguinte opção:

#### **SPNONE**

Nenhuma opção especificada.

Isto é sempre um campo de entrada. O valor inicial deste campo é SPSETF.

### **SPSID (número inteiro assinado de 10 dígitos)**

Este é o identificador de estrutura; o valor deve ser:

#### **SSIDV**

Identificador para configurar a estrutura de opções de propriedade de mensagem.

Isto é sempre um campo de entrada. O valor inicial desse campo é **SPSIDV**.

#### **SPVAKCSI (número inteiro assinado de 10 dígitos).**

O conjunto de caracteres do valor da propriedade a ser configurado se o valor for uma cadeia de caracteres.

Isto é sempre um campo de entrada. O valor inicial desse campo é **CSAPL**.

#### **SPVALENC (número inteiro assinado de 10 dígitos)**

A codificação do valor da propriedade a ser configurado se o valor for numérico.

Isto é sempre um campo de entrada. O valor inicial desse campo é **ENNAT**.

#### **SPVER (número inteiro assinado de 10 dígitos).**

Este é o número da versão da estrutura; o valor deve ser:

##### **SPVER1**

Version-1 configure a estrutura de opções da propriedade de mensagens

A constante a seguir especifica o número da versão atual:

##### **SPVERC**

Versão atual da estrutura de opções de propriedade de mensagem de conjunto.

Isto é sempre um campo de entrada. O valor inicial desse campo é **SPVER1**.

### **Valores iniciais**

<i>Tabela 733. Campos em MQSMPO</i>		
<b>Nome de Campo</b>	<b>Nome da constante</b>	<b>Valor da constante</b>
<i>SPSID</i>	SSIDV	' SMP0 '
<i>SPVER</i>	SPVER1	1
<i>SPOPT</i>	SPNONE	0
<i>SPVALENC</i>	ENNAT	Depende do ambiente
<i>SPVALCSI</i>	CSAPL	-3

### **Declaração RPG**

```
D* MQSMPO Structure
D*
D*
D* Structure identifier
D  SPSID          1      4  INZ('SMP0')
D*
D* Structure version number
D  SPVER          5      8I 0 INZ(1)
D*
** Options that control the action of
D* MQSETMP
D  SPOPT          9     12I 0 INZ(0)
D*
D* Encoding of Value
D  SPVALENC      13     16I 0 INZ(273)
D*
D* Character set identifier of Value
D  SPVALCSI     17     20I 0 INZ(-3)
```

A estrutura MQSRO permite que o aplicativo especifique opções que controlam como uma solicitação de subscrição é feita

## Visão Geral

**Finalidade:** A estrutura é um parâmetro de entrada/saída na chamada MQSUBRQ.

**Versão:** a versão atual do MQSRO é SRVER1..

- “Campos” na página 1274
- “Valores iniciais” na página 1275
- “Declaração RPG” na página 1275

## Campos

A estrutura MQSRO contém os seguintes campos; os campos são descritos em **ordem alfabética**:

### **SRNMP (número inteiro assinado de 10 dígitos)**

Este é um campo de saída, retornado para o aplicativo para indicar o número de publicações enviadas para a fila de assinaturas como resultado dessa chamada. Embora esse número de publicações tenha sido enviado como resultado dessa chamada, não há garantia de que essas muitas mensagens estarão disponíveis para o aplicativo obter, especialmente se forem mensagens não persistentes.

Pode haver mais de uma publicação se o tópico inscrito continha um curinga. Se nenhum curinga estava presente na sequência de tópicos quando a assinatura representada por *HSUB* foi criada, no máximo uma publicação será enviada como resultado dessa chamada..

### **SROPT (número inteiro assinado de 10 dígitos)**

Uma das opções a seguir deve ser especificada: Apenas uma opção pode ser especificada

**Outras opções:** A seguinte opção controla o que acontece quando o gerenciador de filas está em quiesce:

#### **SRFIQ**

A chamada MQSUBRQ falhará se o gerenciador de fila estiver no estado de quiesce

**Opção padrão:** Se a opção descrita anteriormente não for necessária a opção a seguir deve ser usada:

#### **SRNONE**

Use esse valor para indicar que nenhuma outra opção foi especificada. Todas as opções assumem seus valores-padrão.

O SRNONE ajuda a documentação do programa. Embora não seja pretendido que essa opção seja usada com qualquer outra, porque seu valor é zero, esse uso não pode ser detectado..

### **SRSID (cadeia de caracteres de 4 bytes)**

Este é o identificador de estrutura; o valor deve ser:

#### **SRSIDV**

Identificador para a estrutura SROPT de Solicitação de Assinatura

Isto é sempre um campo de entrada. O valor inicial desse campo é SRSIDV.

### **SRVER (número inteiro assinado de 10 dígitos)**

Este é o número da versão da estrutura; o valor deve ser:

#### **SRVER1**

Version-1 Estrutura de Opções de Solicitação de Assinatura.

A constante a seguir especifica o número da versão atual:

## SRVERC

Versão atual da estrutura de Opções de Solicitação de Assinatura

Isto é sempre um campo de entrada. O valor inicial desse campo é SRVER1.

## Valores iniciais

Tabela 734. Campos em MQSRO		
Nome de Campo	Nome da constante	Valor da constante
SRSID	SRSIDV	' SRO↵ '
SRVER	SRVER1	1
SROPT	SRNONE	0
SRNMP	Nenhum	0

**Notas:**

1. O símbolo ↵ representa um único caractere em branco.
2. O valor Sequência nula ou espaços em branco indica a sequência nula em C e caracteres em branco em outras linguagens de programação.

## Declaração RPG

```
D*..1.....2.....3.....4.....5.....6.....7..
D* MQSRO Structure
D*
D* Structure identifier
D SRSID 1 4
D* Structure version number
D SRVER 5 8I 0
D* Options that control the action of MQSUBRQ
D SROPT 9 12I 0
D* Number of publications sent
D SRNMP 13 16I 0
```

## IBM i MQSTS (Estrutura de Relatório de Status) no IBM i

A estrutura MQSTS descreve os dados na estrutura de status retornado pelo comando MQSTAT.

### Visão Geral

**Codificação e conjunto de caracteres:** os dados de caractere em MQSTS estão no conjunto de caracteres do gerenciador de filas locais; isso é fornecido pelo atributo do gerenciador de filas *CodedCharSetId* Os dados numéricos em MQSTS estão na codificação da máquina nativa; isso é fornecido por *ENNAT*

**Uso:** O comando MQSTAT é usado para recuperar as informações de status Essas informações são retornadas na estrutura MQSTS. Para obter informações sobre o MQSTAT, consulte [“MQSTAT \(recuperar informações de status\) em IBM i” na página 1407](#)

- [“Campos” na página 1275](#)
- [“Valores iniciais” na página 1279](#)
- [“Declaração RPG” na página 1279](#)

### Campos

A estrutura MQSTS contém os campos a seguir; os campos são descritos em **ordem alfabética**:

**STSCC (número inteiro assinado de 10 dígitos)**

Este é o código de conclusão resultante do primeiro erro relatado na estrutura MQSTS..

Este é sempre um campo de saída. O valor inicial desse campo é CCOK.

**STSFCC (número inteiro assinado de 10 dígitos)**

Este é o número de chamadas de colocação assíncronas que falharam

Esse é um campo de saída. O valor inicial desse campo é 0.

**STSOBJN (sequência de caracteres de 48 bytes)**

É o nome local do objeto envolvido na primeira falha.

Esse é um campo de saída. O valor inicial desse campo é 48 caracteres em branco.

**STSOQMGR (sequência de caracteres de 48 bytes).**

Este é o nome do gerenciador de filas no qual o objeto *STSOBJN* está definido. Um nome que está inteiramente em branco até o primeiro caractere nulo ou o final do campo denota o gerenciador de fila ao qual o aplicativo está conectado (o gerenciador de filas locais)...

Esse é um campo de saída. O valor inicial desse campo é 48 caracteres em branco.

**STS00 (número inteiro assinado de 10 dígitos)**

O STS00 usado para abrir o objeto que está sendo relatado. Presente apenas na versão 2 de MQSTS ou superior.

O valor de STS00 depende do valor do parâmetro MQSTAT **STYPE** .

**STATAPT**

Zero.

**STATREC**

Zero.

**STATRER**

O STS00 usado quando a falha ocorreu.. A razão para a falha é relatada nos campos *STSCC* e *STSRC* na estrutura MQSTS

STS00 é um campo de saída.. Seu valor inicial é zero.

**STSOS (MQCHARV)**

Nome do objeto longo do objeto com falha que está sendo relatado. Presente apenas na versão 2 de MQSTS ou superior.

STSOS é um campo MQCHARV com um comprimento máximo de 10240. Consulte [MQCHARV](#) , para obter uma descrição de como usar a estrutura MQCHARV

A interpretação de STSOS depende do valor do parâmetro MQSTAT **STYPE** .

**STATAPT**

Este é o nome do objeto longo da fila ou do tópico usado na operação MQPUT , que falhou

**STATREC**

Cadeia de comprimento zero

**STATRER**

Este é o nome longo do objeto que causou a falha da reconexão.

STSOS é um campo de saída.. Seu valor inicial é uma cadeia de comprimento zero.

**STSOT (número inteiro assinado de 10 dígitos)**

O tipo de objeto que está sendo nomeado em *ObjectName* Os valores possíveis são:



**OTALSQ**

Fila de alias.

**OTLOCQ**

Fila local.

**OTMODQ**

Fila modelo.

**OTQ**

Fila.

**OTREMQ**

Fila remota.

**OTTOP**

:NONE.

Este é sempre um campo de saída O valor inicial deste campo é OTQ.

**STSRC (número inteiro assinado de 10 dígitos)**

Este é o código de razão resultante do primeiro erro relatado na estrutura MQSTS

Este é sempre um campo de saída O valor inicial deste campo é RCNONE..

**STSR OBJN (sequência de caracteres de 48 bytes).**

Esse é o nome da fila de destino nomeada em *STSR OBJN* após o gerenciador de fila local resolver o nome.. O nome retornado é o nome de uma fila existente no gerenciador de filas identificado por *STSRQMGR*.

Um valor não em branco será retornado apenas se o objeto for uma única fila aberta para procura, entrada ou saída (ou qualquer combinação). Se o objeto aberto for qualquer um dos seguintes, *STSR OBJN* será configurado como espaços em branco:

- Um tópico
- Uma fila, mas não aberta para procura, entrada ou saída

Esse é um campo de saída. O valor inicial desse campo é 48 caracteres em branco.

**STSRQMGR (sequência de caracteres de 48 bytes);**

Esse é o nome do gerenciador de fila de destino após o gerenciador de fila local resolver o nome.. O nome retornado é o nome do gerenciador de filas que possui a fila identificada por *STSR OBJN*.. *STSRQMGR* pode ser o nome do gerenciador de fila local..

Se *STSR OBJN* for uma fila compartilhada pertencente ao grupo de filas compartilhadas ao qual o gerenciador de filas locais pertence, *STSRQMGR* será o nome do grupo de filas compartilhadas. Se a fila for de propriedade de algum outro grupo de filas compartilhadas, o *STSR OBJN* poderá ser o nome do grupo de filas compartilhadas ou o nome de um gerenciador de filas que é um membro do grupo de filas compartilhadas (a natureza do valor retornado é determinada pelas definições de fila que existem no gerenciador de fila local).

Um valor não em branco será retornado apenas se o objeto for uma única fila aberta para procura, entrada ou saída (ou qualquer combinação). Se o objeto aberto for qualquer um dos seguintes, *STSRQMGR* será configurado como espaços em branco:

- Um tópico
- Uma fila, mas não aberta para procura, entrada ou saída
- Uma fila de clusters com OOBNDN especificado (ou com OOBNDQ em vigor quando o atributo da fila **DefBind** possui o valor OOBNDN)

Esse é um campo de saída. O valor inicial desse campo é 48 caracteres em branco.

**STSSC (número inteiro assinado de 10 dígitos)**

Este é o número de chamadas put assíncronas bem-sucedidas.

Esse é um campo de saída. O valor inicial desse campo é 0.

**STSSID (cadeia de caracteres de 4 bytes).**

Este é o identificador de estrutura. O valor deve ser:.

**STSSID**

Identificador para a estrutura de relatório de status

O valor inicial deste campo é STSSID.

**STSSO (número inteiro assinado de 10 dígitos)**

O STSSO usado para abrir a assinatura com falha.. Presente apenas na versão 2 de MQSTS ou superior.

A interpretação de STSSO depende do valor do parâmetro MQSTAT **STYPE** .

**STATAPT**

Zero.

**STATREC**

Zero.

**STATRER**

O STSSO usado quando a falha ocorreu.. A razão para a falha é relatada nos campos STSCC e STSRC na estrutura MQSTS Se a falha não estiver relacionada à assinatura para um tópico, o valor retornado será zero.

STSSO é um campo de saída.. Seu valor inicial é zero.

**STSSUN (MQCHARV)**

O nome da assinatura com falha.. Presente apenas na versão 2 de MQSTS ou superior.

STSSUN é um campo MQCHARV com um comprimento máximo de 10240. Consulte [MQCHARV](#) , para obter uma descrição de como usar a estrutura MQCHARV

A interpretação de STSSUN depende do valor do parâmetro MQSTAT **STYPE** .

**STATAPT**

Sequência de comprimento zero.

**STATREC**

Sequência de comprimento zero.

**STATRER**

O nome da assinatura que causou a falha de reconexão. Se nenhum nome de assinatura estiver disponível ou a falha não estiver relacionada a uma assinatura, esta será uma sequência de comprimento zero.

STSSUN é um campo de saída.. Seu valor inicial é uma cadeia de comprimento zero.

**STSVR (número inteiro assinado de 10 dígitos)**

Este é o número da versão da estrutura O valor deve ser:.

**STSVR1**

Número da versão para a estrutura de relatório de status

A constante a seguir especifica o número da versão atual:

**STSVRC**

Versão atual da estrutura de relatório de status

O valor inicial desse campo é STSVR1.

## STSWC (número inteiro assinado de 10 dígitos)

Este é o número de chamadas put assíncronas concluídas com um aviso.

Esse é um campo de saída. O valor inicial desse campo é 0.

## Valores iniciais

Tabela 735. Campos em MQSTS		
Nome de Campo	Nome da constante	Valor da constante
STSSID	STSID	
STSVR	STSVRC	STSVR1
STSCC	CCOK	0
STSRC	RCNONE	0
STSSC	Nenhum	0
STSWC	Nenhum	0
STSF	Nenhum	0
STSOT	Nenhum	0
STSOBJN	Nenhum	Espaços em branco
STSOQMGR	Nenhum	Espaços em branco
STSR	Nenhum	Espaços em branco
STSRQMGR	Nenhum	Espaços em branco
STSS	Nomes e valores conforme definidos para MQCHARV	
STSSUN	Nomes e valores conforme definidos para MQCHARV	
STSS0	Nenhum	0
STSS0	Nenhum	0

## Declaração RPG

```
D*..1.....2.....3.....4.....5.....6.....7..
D* MQSTS Structure
D*
D* Structure identifier
D STSSID 1 4
D* Structure version number
D STSVR 5 8I 0
D* Completion code
D STSCC 9 12I 0
D* Reason code
D STSRC 13 16I 0
D* Success count
D STSSC 17 20I 0
D* Warning count
D STSWC 21 24I 0
D* Failure count
D STSF 25 28I 0
D* Object type
D STSOT 29 32I 0
D* Object name
D STSOBJN 33 80
D* Object queue manager
D STSOQMGR 81 128
```

```

D* Resolved object name
D STSROBJN          129    176
D* Resolved object queue manager name
D STSRQMGR          177    224
D* Ver:1 **
D* Failing object long name
D* Address of variable length string
D STSOSCHRP         225    240*
D* Offset of variable length string
D STSOSCHRO         241    244I 0
D* Size of buffer
D STSOSVSBS         245    248I 0
D* Length of variable length string
D STSOSCHRL         249    252I 0
D* CCSID of variable length string
D STSOSCHRC         253    256I 0
D* Failing subscription name
D* Address of variable length string
D STSSUNCHRP        257    272*
D* Offset of variable length string
D STSSUNCHRO        273    276I 0
D* Size of buffer
D STSSUNVSBS        277    280I 0
D* Length of variable length string
D STSSUNCHRL        281    284I 0
D* CCSID of variable length string
D STSSUNCHRC        285    288I 0
D* Failing open options
D STS00             289    292I 0
D* Failing subscription options
D STSS0             293    296I 0
D* Ver:2 **

```

## MQTM-Mensagem do acionador

A estrutura MQTM descreve os dados na mensagem do acionador que é enviada pelo gerenciador de fila para um aplicativo de monitor acionador quando um evento acionador ocorre para uma fila

### Visão Geral

**Propósito:** Essa estrutura faz parte da IBM MQ Trigger Monitor Interface (TMI), que é uma das interfaces de estrutura do IBM MQ

**Nome do formato:** FMTM

**Conjunto de caracteres e codificação:** os dados de caractere no MQTM estão no conjunto de caracteres do gerenciador de filas que gera o MQTM. Os dados numéricos no MQTM estão na codificação de máquina do gerenciador de filas que gera o MQTM

O conjunto de caracteres e a codificação do MQTM são fornecidos pelos campos *MDCSI* e *MDENC* em:

- O MQMD (se a estrutura MQTM estiver no início dos dados da mensagem) ou
- A estrutura do cabeçalho que precede a estrutura MQTM (todos os outros casos).

**Uso:** Um aplicativo de monitor acionador pode precisar transmitir algumas ou todas as informações na mensagem do acionador para o aplicativo que é iniciado pelo aplicativo de monitor acionador. As informações que podem ser necessárias para o aplicativo iniciado incluem *TMQN*, *TMTDe* *TMUD* O aplicativo de monitor acionador pode transmitir a estrutura MQTM diretamente para o aplicativo iniciado ou transmitir uma estrutura MQTMC2, dependendo do que é permitido pelo ambiente e conveniente para o aplicativo iniciado. Para obter informações sobre MQTMC2, consulte [“MQTMC2 \(formato de 2 caracteres da mensagem do acionador\) em IBM i” na página 1284](#).

- No IBM i, o aplicativo do monitor acionador fornecido com IBM MQ passa uma estrutura MQTMC2 para o aplicativo iniciado.

Para obter informações sobre acionadores, consulte [Pré-requisitos para acionamento..](#)

- [“MQMD para uma mensagem do acionador” na página 1281](#)
- [“Campos” na página 1281](#)
- [“Valores iniciais” na página 1284](#)

- [“Declaração RPG” na página 1284](#)

## MQMD para uma mensagem do acionador

Tabela 736. Configurações para os campos no MQMD de uma mensagem acionador gerada pelo gerenciador de filas

Campo no MQMD	Valor Usado
MDSID	MDSIDV
MDVER	MDVER1
MDREP	RONONE
MDMT	MTDGRM
MDEXP	EIULIM
MDFB	FBNONE
MDENC	ENNAT
MDCSI	Atributo <b>CodedCharSetId</b> do gerenciador de filas
MDFMT	FMTM
MDPRI	Atributo <b>DefPriority</b> da fila de inicialização
MDPER	PENPER
MDMID	Um valor exclusivo
MDCID	CINONE
MDBOC	0
MDRQ	Espaços em branco
MDRM	Nome do gerenciador de filas
MDUID	Espaços em branco
MDACC	ACNONE
MDAID	Espaços em branco
MDPAT	ATQM ou conforme apropriado para o agente do canal de mensagens
MDPAN	Primeiros 28 bytes do nome do gerenciador de filas
MDPD	Data quando a mensagem do acionador é enviada
MDPT	Horário quando a mensagem do acionador é enviada
MDAOD	Espaços em branco

Um aplicativo que gera uma mensagem do acionador é recomendado para configurar valores semelhantes, exceto o seguinte:

- O campo *MDPRI* pode ser configurado como PRQDEF (o gerenciador de filas mudará isso para a prioridade padrão para a fila de inicialização quando a mensagem for colocada).
- O campo *MDRM* pode ser definido como espaços em branco (o gerenciador de filas alterará isso para o nome do gerenciador de fila local quando a mensagem for colocada)...
- Os campos de contexto devem ser configurados conforme apropriado para o aplicativo

### Campos

A estrutura MQTM contém os seguintes campos; os campos são descritos em **ordem alfabética**:

### **TMAI (sequência de caracteres de 256 bytes)**

O identificador do aplicativo.

Essa é uma sequência de caracteres que identifica o aplicativo a ser iniciado e é usada pelo aplicativo acionador-monitor que recebe a mensagem do acionador O gerenciador de filas inicializa esse campo com o valor do atributo **App1Id** do objeto de processo identificado pelo campo *TMPN* ; consulte [“Atributos para definições de processo em IBM i” na página 1447](#) para obter detalhes desse atributo. O conteúdo desses dados não é significativo para o gerenciador da fila.

O significado de *TMAI* é determinado pelo aplicativo do monitor acionador O monitor acionador fornecido pelo IBM MQ requer que *TMAI* seja o nome de um programa executável

O comprimento deste campo é dado por LNPROA. O valor inicial desse campo é 256 caracteres em branco.

### **TMAT (número inteiro assinado de 10 dígitos)**

Tipo de Aplicativo.

Isso identifica a natureza do programa a ser iniciado e é usado pelo aplicativo monitor acionador que recebe a mensagem do acionador. O gerenciador de filas inicializa esse campo com o valor do atributo **App1Type** do objeto de processo identificado pelo campo *TMPN* ; consulte [“Atributos para definições de processo em IBM i” na página 1447](#) para obter detalhes desse atributo. O conteúdo desses dados não é significativo para o gerenciador da fila.

*TMAT* pode ter um dos seguintes valores padrão. Os tipos definidos pelo usuário também podem ser usados, mas devem ser restritos a valores no intervalo de ATUFST a ATULST:

#### **àsCICS**

Transação CICS .

#### **ATVSE**

Transação CICS/VSE .

#### **AT400**

IBM i .

#### **ATUFST**

Valor mais baixo para o tipo de aplicativo definido pelo usuário.

#### **ATULST**

Valor mais alto para o tipo de aplicativo definido pelo usuário.

O valor inicial desse campo é 0.

### **TMED (cadeia de caracteres de 128 bytes).**

Dados do ambiente

Esta é uma sequência de caracteres que contém informações relacionadas ao ambiente pertencentes ao aplicativo a ser iniciado e é usada pelo aplicativo acionador-monitor que recebe a mensagem do acionador O gerenciador de filas inicializa esse campo com o valor do atributo **EnvData** do objeto de processo identificado pelo campo *TMPN* ; consulte [“Atributos para definições de processo em IBM i” na página 1447](#) para obter detalhes desse atributo. O conteúdo desses dados não é significativo para o gerenciador da fila.

O comprimento deste campo é fornecido por LNPROE O valor inicial deste campo é 128 caracteres em branco.

### **TMPN (cadeia de caracteres de 48 bytes)**

Nome do objeto de processo..

Este é o nome do objeto do processo do gerenciador de filas especificado para a fila acionada e pode ser usado pelo aplicativo do monitor acionador que recebe a mensagem do acionador O gerenciador

de filas inicializa esse campo com o valor do atributo **ProcessName** da fila identificada pelo campo *TMQN* ; consulte [“Atributos para filas” na página 1416](#) para obter detalhes desse atributo.

Os nomes que são menores que o comprimento definido do campo são sempre preenchidos à direita com espaços em branco; eles não são terminados prematuramente por um caractere nulo.

O comprimento deste campo é fornecido por LNPRON (LNPRON). O valor inicial desse campo é 48 caracteres em branco.

#### **TMQN (sequência de caracteres de 48 bytes).**

Nome da fila acionada..

Este é o nome da fila para a qual ocorreu um evento acionador e é usado pelo aplicativo iniciado pelo aplicativo do monitor acionador O gerenciador de filas inicializa esse campo com o valor do atributo **QName** da fila acionada; consulte [“Atributos para filas” na página 1416](#) para obter detalhes desse atributo.

Os nomes que são mais curtos que o comprimento definido do campo são preenchidos à direita com espaços em branco; eles não são encerrados prematuramente por um caractere nulo.

O comprimento deste campo é fornecido por LNQN (LNQN). O valor inicial desse campo é 48 caracteres em branco.

#### **TMSID (sequência de caracteres de 4 bytes)**

Identificador de estruturação.

O valor deve ser:.

##### **TMSIDV**

Identificador para a estrutura da mensagem do acionador

O valor inicial deste campo é TMSIDV.

#### **TMTD (sequência de caracteres de 64 bytes).**

Dados do acionador.

Esses são dados de formato livre para uso pelo aplicativo de monitor acionador que recebe a mensagem do acionador. O gerenciador de filas inicializa esse campo com o valor do atributo **TriggerData** da fila identificada pelo campo *TMQN* ; consulte [“Atributos para filas” na página 1416](#) para obter detalhes desse atributo. O conteúdo desses dados não é significativo para o gerenciador da fila.

O comprimento deste campo é fornecido pelo LNTRGD O valor inicial desse campo é de 64 caracteres em branco.

#### **TMUD (cadeia de caracteres de 128 bytes)**

Dados do usuário.

Essa é uma sequência de caracteres que contém informações do usuário relevantes para o aplicativo a ser iniciado e é usada pelo aplicativo monitor acionador que recebe a mensagem do acionador. O gerenciador de filas inicializa esse campo com o valor do atributo **UserData** do objeto de processo identificado pelo campo *TMPN* ; consulte [“Atributos para definições de processo em IBM i” na página 1447](#) para obter detalhes desse atributo. O conteúdo desses dados não é significativo para o gerenciador da fila.

O comprimento deste campo é fornecido por LNPROU O valor inicial deste campo é 128 caracteres em branco.

#### **TMVER (número inteiro assinado de 10 dígitos)**

Número de versão da estrutura.

O valor deve ser:.

##### **TMVER1**

Número da versão para a estrutura da mensagem do acionador

A constante a seguir especifica o número da versão atual:

### TMVERC

A versão atual da estrutura da mensagem do acionador

O valor inicial desse campo é TMVER1.

## Valores iniciais

Tabela 737. Campos em MQTM		
Nome de Campo	Nome da constante	Valor da constante
TMSID	TMSIDV	'TM↵↵'
TMVER	TMVER1	1
TMQN	Nenhum	Espaços em branco
TMPN	Nenhum	Espaços em branco
TMTD	Nenhum	Espaços em branco
TMAT	Nenhum	0
TMAI	Nenhum	Espaços em branco
TMED	Nenhum	Espaços em branco
TMUD	Nenhum	Espaços em branco
<b>Notas:</b>		
1. O símbolo ↵ representa um único caractere em branco.		

## Declaração RPG

```

D*..1.....2.....3.....4.....5.....6.....7..
D*
D* MQTM Structure
D*
D* Structure identifier
D TMSID          1      4  INZ('TM ')
D* Structure version number
D TMVER          5      8I 0 INZ(1)
D* Name of triggered queue
D TMQN          9      56  INZ
D* Name of process object
D TMPN         57      104  INZ
D* Trigger data
D TMTD        105      168  INZ
D* Application type
D TMAT        169      172I 0 INZ(0)
D* Application identifier
D TMAI        173      428  INZ
D* Environment data
D TMED        429      556  INZ
D* User data
D TMUD        557      684  INZ

```

## IBM i MQTMC2 (formato de 2 caracteres da mensagem do acionador) em IBM i

Quando um aplicativo de monitor acionador recupera uma mensagem do acionador (MQTM) de uma fila de inicialização, o monitor acionador pode precisar transmitir algumas ou todas as informações na mensagem do acionador para o aplicativo que é iniciado pelo monitor acionador.



## Visão Geral

**Propósito:** as informações que podem ser necessárias pelo aplicativo iniciado incluem *TC2QN*, *TC2TDe* e *TC2UD*. O aplicativo do monitor acionador pode transmitir a estrutura MQTM diretamente para o aplicativo iniciado ou transmitir uma estrutura MQTMC2 em vez disso, dependendo do que é permitido pelo ambiente e conveniente para o aplicativo iniciado.

Essa estrutura faz parte da IBM MQ Trigger Monitor Interface (TMI), que é uma das interfaces de estrutura IBM MQ .

**Codificação e conjunto de caracteres:** os dados de caracteres em MQTMC2 estão no conjunto de caracteres do gerenciador de filas locais; isso é fornecido pelo atributo do gerenciador de filas **CodedCharSetId** .

**Uso:** A estrutura MQTMC2 é semelhante ao formato da estrutura MQTM. A diferença é que os campos não de caracteres no MQTM são alterados no MQTMC2 para campos de caracteres do mesmo comprimento e o nome do gerenciador de filas é incluído no final da estrutura.

- No IBM i, o aplicativo de monitor acionador fornecido com IBM MQ passa uma estrutura MQTMC2 para o aplicativo iniciado.
- [“Campos” na página 1285](#)
- [“Valores iniciais” na página 1286](#)
- [“Declaração RPG” na página 1286](#)

## Campos

A estrutura MQTMC2 contém os campos a seguir; os campos são descritos em **ordem alfabética**:

### **TC2AI (sequência de caracteres de 256 bytes)**

O identificador do aplicativo.

Consulte o campo *TMAI* na estrutura do MQTM

### **TC2AT (sequência de caracteres de 4 bytes)**

Tipo de Aplicativo.

Esse campo sempre contém espaços em branco, independentemente do valor no campo *TMAT* na estrutura MQTM da mensagem do acionador original.

### **TC2ED (sequência de caracteres de 128 bytes).**

Dados do ambiente

Consulte o campo *TMED* na estrutura do MQTM

### **TC2PN (sequência de caracteres de 48 bytes)**

Nome do objeto de processo..

Consulte o campo *TMPN* na estrutura do MQTM

### **TC2QMN (sequência de caracteres de 48 bytes).**

Nome do gerenciador de filas.

Este é o nome do gerenciador de filas no qual o evento acionador ocorreu

### **TC2QN (sequência de caracteres de 48 bytes).**

Nome da fila acionada..

Consulte o campo *TMQN* na estrutura do MQTM

### **TC2SID (sequência de caracteres de 4 bytes)**

Identificador de estruturação.

O valor deve ser:.

**TCSIDV**

Identificador para a estrutura da mensagem do acionador (formato de caractere)

**TC2TD (sequência de caracteres de 64 bytes).**

Dados do acionador.

Consulte o campo *TMTD* na estrutura do MQTM

**TC2UD (sequência de caracteres de 128 bytes)**

Dados do usuário.

Consulte o campo *TMUD* na estrutura do MQTM

**TC2VER (sequência de caracteres de 4 bytes)**

Número de versão da estrutura.

O valor deve ser:.

**TCVER2**

Estrutura da mensagem do acionador versão 2 (formato de caractere).

A constante a seguir especifica o número da versão atual:

**TCVERC**

Versão atual da estrutura da mensagem do acionador (formato de caractere).

**Valores iniciais**

Tabela 738. Campos em MQTMC2		
Nome de Campo	Nome da constante	Valor da constante
TC2SID	TCSIDV	'TMC¬'
TC2VER	TCVER2	'¬¬¬2'
TC2QN	Nenhum	Espaços em branco
TC2PN	Nenhum	Espaços em branco
TC2TD	Nenhum	Espaços em branco
TC2AT	Nenhum	Espaços em branco
TC2AI	Nenhum	Espaços em branco
TC2ED	Nenhum	Espaços em branco
TC2UD	Nenhum	Espaços em branco
TC2QMN	Nenhum	Espaços em branco
<b>Notas:</b>		
1. O símbolo ¬ representa um único caractere em branco.		

**Declaração RPG**

```

D*..1.....2.....3.....4.....5.....6.....7..
D* MQTMC2 Structure
D*
D* Structure identifier
D TC2SID           1      4
D* Structure version number
D TC2VER           5      8
    
```

D* Name of triggered queue		
D TC2QN	9	56
D* Name of process object		
D TC2PN	57	104
D* Trigger data		
D TC2TD	105	168
D* Application type		
D TC2AT	169	172
D* Application identifier		
D TC2AI	173	428
D* Environment data		
D TC2ED	429	556
D* User data		
D TC2UD	557	684
D* Queue manager name		
D TC2QMN	685	732

IBM i

## MQWIH (cabeçalho de informações de trabalho) no IBM i

A estrutura MQWIH descreve as informações que devem estar presentes no início de uma mensagem que deve ser manipulado pelo gerenciador de cargas de trabalho do z/OS

### Visão Geral

**Nome do formato:** FMWIH

**Conjunto de caracteres e codificação:** os campos na estrutura MQWIH estão no conjunto de caracteres e codificação fornecidos pelos campos *MDCSI* e *MDENC* na estrutura do cabeçalho que precede MQWIH ou por esses campos na estrutura MQMD se o MQWIH estiver no início dos dados da mensagem do aplicativo.

O conjunto de caracteres deve ser um que tenha caracteres de byte único para os caracteres válidos em nomes de filas.

**Uso:** Se uma mensagem deve ser processada pelo gerenciador de carga de trabalho do z/OS , a mensagem deve começar com uma estrutura MQWIH.

- [“Campos” na página 1287](#)
- [“Valores iniciais” na página 1289](#)
- [“Declaração RPG” na página 1290](#)

### Campos

A estrutura MQWIH contém os seguintes campos; os campos são descritos em **ordem alfabética**:

#### WICSI (número inteiro assinado de 10 dígitos)

Identificador do conjunto de caracteres de dados que segue MQWIH.

Isso especifica o identificador do conjunto de caracteres dos dados que seguem a estrutura MQWIH; ele não se aplica aos dados de caracteres na própria estrutura MQWIH...

Na chamada MQPUT ou MQPUT1 , o aplicativo deve configurar esse campo para o valor apropriado aos dados. O seguinte valor especial pode ser usado:

#### CSINHT

Herde o identificador do conjunto de caracteres desta estrutura..

Os dados de caractere nos dados *após* essa estrutura estão no mesmo conjunto de caracteres que essa estrutura

O gerenciador de filas altera esse valor na estrutura enviada na mensagem para o identificador do conjunto de caractere real da estrutura.. Se nenhum erro ocorrer, o valor CSINHT não será retornado pela chamada MQGET.

CSINHT não poderá ser usado se o valor do campo *MDPAT* em MQMD for ATBRKR.

O valor inicial desse campo é CSUNDF.

**WIENC (número inteiro assinado de 10 dígitos)**

Codificação numérica de dados após MQWIH.

Isso especifica a codificação numérica dos dados que seguem a estrutura MQWIH; ela não se aplica a dados numéricos na própria estrutura MQWIH

Na chamada MQPUT ou MQPUT1 , o aplicativo deve configurar esse campo para o valor apropriado aos dados.

O valor inicial desse campo é 0.

**WIFLG (número inteiro assinado de 10 dígitos)**

Sinalizadores

O valor deve ser:.

**WINONE**

Sem bandeiras.

O valor inicial deste campo é WINONE.

**WIFMT (cadeia de caracteres de 8 bytes)**

Formatar o nome dos dados que seguem MQWIH

Isso especifica o nome do formato dos dados que seguem a estrutura MQWIH

Na chamada MQPUT ou MQPUT1 , o aplicativo deve configurar esse campo para o valor apropriado aos dados. As regras para codificação desse campo são as mesmas que aquelas para o campo *MDFMT* no MQMD.

O comprimento deste campo é fornecido pelo LNFMT O valor inicial deste campo é FMNONE.

**WILEN (número inteiro assinado de 10 dígitos)**

Comprimento da estrutura MQWIH..

O valor deve ser:.

**WILEN1**

Comprimento da estrutura do cabeçalho de informações de trabalho version-1 .

A constante a seguir especifica o comprimento da versão atual:

**WILENC**

Comprimento da versão atual da estrutura do cabeçalho de informações de trabalho

O valor inicial desse campo é WILEN1.

**WIRSV (sequência de caracteres de 32 bytes)**

Reservado.

Este é um campo reservado; ele deve estar em branco

**WISID (sequência de caracteres de 4 bytes)**

Identificador de estruturação.

O valor deve ser:.

**WISIDV**

Identificador para estrutura do cabeçalho de informações de trabalho.

O valor inicial deste campo é WISIDV.

**WISNM (cadeia de caracteres de 32 bytes)**

Nome do serviço.

É o nome do serviço que deve processar a mensagem.

O comprimento deste campo é fornecido por LNSVNM O valor inicial desse campo é 32 caracteres em branco.

### **WISST (cadeia de caracteres de 8 bytes)**

Nome da etapa de serviço..

Este é o nome da etapa de *WISNM* à qual a mensagem se relaciona

O comprimento deste campo é fornecido pelo LNSVST. O valor inicial deste campo é 8 caracteres em branco.

### **WITOK (cadeia de bits de 16 bytes)**

Token de mensagem

Este é um token de mensagem que identifica exclusivamente a mensagem..

Para as chamadas MQPUT e MQPUT1 , esse campo é ignorado. O comprimento deste campo é fornecido por LNMTOK O valor inicial deste campo é MTKNON.

### **WIVER (número inteiro assinado de 10 dígitos)**

Número de versão da estrutura.

O valor deve ser:.

#### **WIVER1**

Version-1 estrutura do cabeçalho de informações de trabalho.

A constante a seguir especifica o número da versão atual:

#### **WIVERC**

Versão atual da estrutura do cabeçalho de informações de trabalho

O valor inicial desse campo é WIVER1.

## **Valores iniciais**

<i>Tabela 739. Campos em MQWIH</i>		
<b>Nome de Campo</b>	<b>Nome da constante</b>	<b>Valor da constante</b>
<i>WISID</i>	WISIDV	'WIH~'
<i>WIVER</i>	WIVER1	1
<i>WILEN</i>	WILEN1	120
<i>WIENC</i>	Nenhum	0
<i>WICSI</i>	CSUNDF	0
<i>WIFMT</i>	FMNONE	Espaços em branco
<i>WIFLG</i>	WINONE	0
<i>WISNM</i>	Nenhum	Espaços em branco
<i>WISST</i>	Nenhum	Espaços em branco
<i>WITOK</i>	MTKNON	Nulos
<i>WIRSV</i>	Nenhum	Espaços em branco
<b>Notas:</b>		
1. O símbolo ~ representa um único caractere em branco.		

## Declaração RPG

```
D*..1.....2.....3.....4.....5.....6.....7..
D*
D* MQWIH Structure
D*
D* Structure identifier
D WISID          1          4    INZ('WIH ')
D* Structure version number
D WIVER          5          8I 0 INZ(1)
D* Length of MQWIH structure
D WILEN          9          12I 0 INZ(120)
D* Numeric encoding of data that followsMQWIH
D WIENC          13         16I 0 INZ(0)
D* Character-set identifier of data thatfollows MQWIH
D WICSI          17         20I 0 INZ(0)
D* Format name of data that followsMQWIH
D WIFMT          21         28    INZ('      ')
D* Flags
D WIFLG          29         32I 0 INZ(0)
D* Service name
D WISNM          33         64    INZ
D* Service step name
D WISST          65         72    INZ
D* Message token
D WITOK          73         88    INZ(X'0000000000000000-
D                                     0000000000000000')
D* Reserved
D WIRSV          89         120   INZ
```



## MQXQH (Transmission-queue header) em IBM i

A estrutura MQXQH descreve as informações prefixadas para os dados da mensagem do aplicativo de mensagens quando elas estão em filas de transmissão.

### Visão Geral

**Propósito:** Uma fila de transmissão é um tipo especial de fila local que contém temporariamente mensagens destinadas a filas remotas (ou seja, destinadas a filas que não pertencem ao gerenciador de filas locais). Uma fila de transmissão é denotada pelo atributo da fila **Usage** com o valor USTRAN.

**Nome do Formato:** FMXQH.

**Conjunto de caracteres e codificação:** os dados em MQXQH devem estar no conjunto de caracteres fornecido pelo atributo do gerenciador de fila do **CodedCharSetId** e a codificação do gerenciador de fila local fornecida pelo ENNAT para a linguagem de programação C...

O conjunto de caracteres e a codificação do MQXQH devem ser configurados nos campos *MDCSI* e *MDENC* em:

- O MQMD separado (se a estrutura MQXQH estiver no início dos dados da mensagem) ou
- A estrutura do cabeçalho que precede a estrutura MQXQH (todos os outros casos).

**Uso:** Uma mensagem que está em uma fila de transmissão tem *dois* descritores de mensagens:

- Um descritor de mensagens é armazenado separadamente dos dados da mensagem; isso é chamado de *descritor de mensagem separado* e é gerado pelo gerenciador de filas quando a mensagem é colocada na fila de transmissão Alguns dos campos no descritor de mensagens separado são copiados do descritor de mensagens fornecido pelo aplicativo na chamada MQPUT ou MQPUT1 .

O descritor de mensagens separado é aquele retornado para o aplicativo no parâmetro **MSGDSC** da chamada MQGET quando a mensagem é removida da fila de transmissão.

- Um segundo descritor de mensagens é armazenado na estrutura MQXQH como parte dos dados da mensagem; isso é chamado de *descritor de mensagens integrado* e é uma cópia do descritor de mensagens que foi fornecido pelo aplicativo na chamada MQPUT ou MQPUT1 (com variações menores).

O descritor de mensagens integrado é sempre um MQMD version-1 . Se a mensagem colocada pelo aplicativo tiver valores não padrão para um ou mais dos campos version-2 no MQMD, uma estrutura

MQMDE seguirá o MQXQH e será seguida pelos dados da mensagem do aplicativo (se houver). O MQMDE é:

- Gerado pelo gerenciador de filas (se o aplicativo usar um MQMD version-2 para colocar a mensagem) ou
- Já presente no início dos dados da mensagem do aplicativo (se o aplicativo usar um MQMD version-1 para colocar a mensagem).

O descritor de mensagens integrado é aquele retornado para o aplicativo no parâmetro **MSGDSC** da chamada MQGET quando a mensagem é removida da fila de destino final.

- [“Campos no Descritor de Mensagens Separado” na página 1291](#)
- [“Campos no descritor de mensagens integrado” na página 1292](#)
- [“Colocando mensagens em filas remotas” na página 1293](#)
- [“Colocando mensagens diretamente em filas de transmissão” na página 1293](#)
- [“Obtendo mensagens de filas de transmissão” na página 1293](#)
- [“Campos” na página 1293](#)
- [“Valores iniciais” na página 1294](#)
- [“Declaração RPG” na página 1295](#)

## Campos no Descritor de Mensagens Separado

Os campos no descritor de mensagem separado são configurados pelo gerenciador de filas, conforme mostrado na lista a seguir: Se o gerenciador de filas não suportar o MQMD version-2, um MQMD version-1 será usado sem perda de função.

Tabela 740. Campos no Descritor de Mensagens Separado e Valores Utilizados

<b>Campo em MQMD separado.</b>	<b>Valor Usado</b>
<i>MDSID</i>	MDSIDV
<i>MDVER</i>	MDVER2
<i>MDREP</i>	Copiado do descritor de mensagens incorporado, mas com os bits identificados por ROAUXM definidos como zero. (Isso impede que uma mensagem de relatório COA ou COD seja gerada quando uma mensagem é colocada ou removida de uma fila de transmissão.)
<i>MDMT</i>	Copiado do descritor de mensagens integrado.
<i>MDEXP</i>	Copiado do descritor de mensagens integrado.
<i>MDFB</i>	Copiado do descritor de mensagens integrado.
<i>MDENC</i>	ENNAT
<i>MDCSI</i>	atributo <b>CodedCharSetId</b> do gerenciador de filas.
<i>MDFMT</i>	FMXQH
<i>MDPRI</i>	Copiado do descritor de mensagens integrado.
<i>MDPER</i>	Copiado do descritor de mensagens integrado.
<i>MDMID</i>	Um novo valor é gerado pelo gerenciador de fila. Esse identificador de mensagem é diferente do <i>MDMID</i> que o gerenciador de filas pode ter gerado para o descritor de mensagem integrado (consulte descrito anteriormente)...
<i>MDCID</i>	O <i>MDMID</i> do descritor de mensagens integrado.
<i>MDBOC</i>	0

Tabela 740. Campos no Descritor de Mensagens Separado e Valores Utilizados (continuação)

<b>Campo em MQMD separado.</b>	<b>Valor Usado</b>
<i>MDRQ</i>	Copiado do descritor de mensagens integrado.
<i>MDRM</i>	Copiado do descritor de mensagens integrado.
<i>MDUID</i>	Copiado do descritor de mensagens integrado.
<i>MDACC</i>	Copiado do descritor de mensagens integrado.
<i>MDAID</i>	Copiado do descritor de mensagens integrado.
<i>MDPAT</i>	ATQM
<i>MDPAN</i>	Os primeiros 28 bytes do nome do gerenciador de filas
<i>MDPD</i>	Data quando a mensagem foi colocada na fila de transmissão.
<i>MDPT</i>	O horário em que a mensagem foi colocada na fila de transmissão
<i>MDAOD</i>	Espaços em branco
<i>MDGID</i>	GINONE
<i>MDSEQ</i>	1
<i>MDOFF</i>	0
<i>MDMFL</i>	MFNONE
<i>MDOLN</i>	OLUNDF

### **Campos no descritor de mensagens integrado**

Os campos no descritor de mensagens integradas têm os mesmos valores que aqueles no parâmetro **MSGDSC** da chamada MQPUT ou MQPUT1, exceto o seguinte:

- O campo *MDVER* sempre tem o valor MDVER1.
- Se o campo *MDPRI* tiver o valor PRQDEF, ele será substituído pelo valor do atributo **DefPriority** da filas.
- Se o campo *MDPER* tiver o valor PEQDEF, ele será substituído pelo valor do atributos **DefPersistence** da fila.
- Se o campo *MDMID* tiver o valor MINONE ou a opção PMNMID tiver sido especificada ou a mensagem for uma mensagem da lista de distribuição, *MDMID* será substituído por um novo identificador de mensagem gerado pelo gerenciador de filas.

Quando uma mensagem da lista de distribuição é dividida em mensagens da lista de distribuição menores colocadas em filas de transmissão diferentes, o campo *MDMID* em cada um dos novos descritores de mensagens integrados é o mesmo que o da mensagem da lista de distribuição original

- Se a opção PMNCID foi especificada, *MDCID* será substituído por um novo identificador de correlação gerado pelo gerenciador de filas.
- Os campos de contexto são configurados conforme indicado pelas opções PM\* especificadas no parâmetro **PMO**; os campos de contexto são:
  - *MDACC*
  - *MDAID*
  - *MDAOD*
  - *MDPAN*
  - *MDPAT*
  - *MDPD*
  - *MDPT*



– MDUID

- Os campos version-2 (se estavam presentes) são removidos do MQMD e movidos para uma estrutura MQMDE, se um ou mais dos campos version-2 tiverem um valor não padrão.

## Colocando mensagens em filas remotas

: Quando um aplicativo coloca uma mensagem em uma fila remota (especificando o nome da fila remota diretamente ou usando uma definição local da fila remota), o gerenciador de fila local:

- Cria uma estrutura MQXQH contendo o descritor de mensagens integrado
- Anexa um MQMDE se um for necessário e ainda não estiver presente
- Anexa os dados da mensagem do aplicativo..
- Coloca a mensagem em uma fila de transmissão apropriada

## Colocando mensagens diretamente em filas de transmissão

Também é possível que um aplicativo coloque uma mensagem diretamente em uma fila de transmissão. Neste caso, o aplicativo deve prefixar os dados da mensagem do aplicativo com uma estrutura MQXQH, e inicializar os campos com valores apropriados. Além disso, o campo *MDFMT* no parâmetro **MSGDSC** da chamada MQPUT ou MQPUT1 deve ter o valor FMXQH.

Os dados de caracteres na estrutura MQXQH criada pelo aplicativo devem estar no conjunto de caracteres do gerenciador de filas locais (definido pelo atributo do gerenciador de filas **CodedCharSetId**) e os dados de número inteiro devem estar na codificação da máquina nativa. Além disso, os dados de caracteres na estrutura MQXQH devem ser preenchidos com espaços em branco para o comprimento definido do campo; os dados não devem ser encerrados prematuramente usando um caractere nulo, porque o gerenciador de fila não converte os caracteres nulos e subsequentes em espaços em branco na estrutura MQXQH...

Observe, no entanto, que o gerenciador de filas não verifica se uma estrutura MQXQH está presente ou se valores válidos foram especificados para os campos

## Obtendo mensagens de filas de transmissão

Os aplicativos que obtêm mensagens de uma fila de transmissão devem processar as informações na estrutura MQXQH de uma maneira apropriada. A presença da estrutura MQXQH no início dos dados da mensagem do aplicativo é indicada pelo valor FMXQH sendo retornado no campo *MDFMT* no parâmetro **MSGDSC** da chamada MQGET. Os valores retornados nos campos *MDCSI* e *MDENC* no parâmetro **MSGDSC** indicam o conjunto de caracteres e a codificação dos caracteres e dados de número inteiro na estrutura MQXQH. O conjunto de caracteres e a codificação dos dados da mensagem do aplicativo são definidos pelos campos *MDCSI* e *MDENC* no descritor de mensagem integrado

## Campos

A estrutura MQXQH contém os seguintes campos; os campos são descritos em **ordem alfabética**:

### XQMD (MQMD1)

Descritor de mensagens original

Esse é o descritor de mensagens integrado e é uma cópia próxima do descritor de mensagens MQMD que foi especificado como o parâmetro **MSGDSC** na chamada MQPUT ou MQPUT1 quando a mensagem foi originalmente colocada na fila remota.

**Nota:** Este é um MQMD version-1 .

Os valores iniciais dos campos nessa estrutura são os mesmos da estrutura MQMD.

### XQRQ (sequência de caracteres de 48 bytes)

Nome da fila de destino..

Este é o nome da fila de mensagens que é o destino eventual aparente para a mensagem (isso pode provar que não será o destino eventual real se, por exemplo, esta fila for definida em *XQRQM* para ser uma definição local de outra fila remota)

Se a mensagem for uma mensagem da lista de distribuição (ou seja, o campo *MDFMT* no descritor de mensagens integrado for *FMDH*), *XQRQ* estará em branco.

O comprimento deste campo é fornecido por *LNQN* (*LNQN*). O valor inicial desse campo é 48 caracteres em branco.

#### **XQRQM (cadeia de caracteres de 48 bytes).**

Nome do gerenciador de filas de destino..

Este é o nome do gerenciador de filas ou do grupo de filas compartilhadas que possui a fila que é o destino eventual aparente para a mensagem

Se a mensagem for uma mensagem da lista de distribuições, *XQRQM* estará em branco

O comprimento deste campo é fornecido por *LNQMN* O valor inicial desse campo é 48 caracteres em branco.

#### **XQSID (sequência de caracteres de 4 bytes).**

Identificador de estruturação.

O valor deve ser:.

##### **XQSIDV**

Identificador para a estrutura do cabeçalho da fila de transmissão

O valor inicial desse campo é *XQSIDV*.

#### **XQVER (número inteiro assinado de 10 dígitos)**

Número de versão da estrutura.

O valor deve ser:.

##### **XQVER1**

Número da versão para a estrutura do cabeçalho da fila de transmissão

A constante a seguir especifica o número da versão atual:

##### **XQVERC**

Versão atual da estrutura do cabeçalho da fila de transmissão..

O valor inicial desse campo é *XQVER1*.

### **Valores iniciais**

<i>Tabela 741. Campos em MQXQH</i>		
<b>Nome de Campo</b>	<b>Nome da constante</b>	<b>Valor da constante</b>
<i>XQSID</i>	<i>XQSIDV</i>	'XQH↵'
<i>XQVER</i>	<i>XQVER1</i>	1
<i>XQRQ</i>	Nenhum	Espaços em branco
<i>XQRQM</i>	Nenhum	Espaços em branco
<i>XQMD</i>	Mesmos nomes e valores que <i>MQMD</i> ; consulte <a href="#">Tabela 709 na página 1192</a>	-
<b>Notas:</b>		
1. O símbolo ↵ representa um único caractere em branco.		

## Declaração RPG

```
D*..1....:....2....:....3....:....4....:....5....:....6....:....7..
D*
D* MQXQH Structure
D*
D* Structure identifier
D XQSID          1          4    INZ('XQH ')
D* Structure version number
D XQVER          5          8I 0  INZ(1)
D* Name of destination queue
D XQRQ           9          56    INZ
D* Name of destination queue manager
D XQRQM          57         104    INZ
D* Original message descriptor
D XQ1SID         105        108    INZ('MD ')
D XQ1VER         109        112I 0  INZ(1)
D XQ1REP         113        116I 0  INZ(0)
D XQ1MT          117        120I 0  INZ(8)
D XQ1EXP         121        124I 0  INZ(-1)
D XQ1FB          125        128I 0  INZ(0)
D XQ1ENC         129        132I 0  INZ(273)
D XQ1CSI         133        136I 0  INZ(0)
D XQ1FMT         137        144    INZ(' ')
D XQ1PRI         145        148I 0  INZ(-1)
D XQ1PER         149        152I 0  INZ(2)
D XQ1MID         153        176    INZ(X'00000000000000-
D                               0000000000000000000000-
D                               00000000000000')
D XQ1CID         177        200    INZ(X'00000000000000-
D                               0000000000000000000000-
D                               000000000000')
D XQ1BOC         201        204I 0  INZ(0)
D XQ1RQ          205        252    INZ
D XQ1RM          253        300    INZ
D XQ1UID         301        312    INZ
D XQ1ACC         313        344    INZ(X'00000000000000-
D                               0000000000000000000000-
D                               000000000000000000-
D                               00000000')
D XQ1AID         345        376    INZ
D XQ1PAT         377        380I 0  INZ(0)
D XQ1PAN         381        408    INZ
D XQ1PD          409        416    INZ
D XQ1PT          417        424    INZ
D XQ1AOD         425        428    INZ
```

IBM i

## Chamadas de função em IBM i .

Use estas informações para aprender sobre as chamadas de função disponíveis na programação IBM i .

### Convenções usadas nas descrições de chamada em IBM i

Para cada chamada, esta coleção de tópicos fornece uma descrição dos parâmetros e do uso da chamada.. Isso é seguido por chamadas típicas da chamada e declarações típicas de seus parâmetros na linguagem de programação RPG.

**Importante:** Ao codificar chamadas API do IBM MQ , deve-se assegurar que todos os parâmetros relevantes (conforme descrito nas seções a seguir) sejam fornecidos. A falha em fazer isso pode produzir resultados imprevisíveis.

A descrição de cada chamada contém as seguintes seções:

#### Nome da chamada

O nome da chamada, seguido por uma breve descrição do propósito da chamada

#### Parâmetros

Para cada parâmetro, o nome é seguido por seu tipo de dados entre parênteses () e sua direção; por exemplo:

*CMPCOD* (número inteiro decimal de 9 dígitos)-saída.

Há mais informações sobre os tipos de dados de estrutura em [“Tipos de dados elementares” na página 1031](#)

A direção do parâmetro pode ser:

#### **Entrada**

Você (programador) deve fornecer esse parâmetro.

#### **Saída**

A chamada retorna esse parâmetro..

#### **Entrada/saída**

Deve-se fornecer esse parâmetro, mas ele é modificado pela chamada

Há também uma breve descrição do propósito do parâmetro, juntamente com uma lista de quaisquer valores que o parâmetro pode assumir.

Os dois últimos parâmetros em cada chamada são um código de conclusão e um código de razão. O código de conclusão indica se a chamada foi concluída com êxito, parcialmente ou não. Informações adicionais sobre o sucesso parcial ou falha da chamada são fornecidas no código de razão.

#### **Observações de Uso**

Informações adicionais sobre a chamada, descrevendo como usá-la e quaisquer restrições sobre seu uso.

#### **Chamada RPG**

Chamada típica da chamada e declaração de seus parâmetros no RPG.

Outras convenções notacionais são:

#### **Constantes**

Nomes de constantes são mostrados em maiúsculas; por exemplo, OOOUT.

#### **Matrizes**

Em algumas chamadas, parâmetros são matrizes de sequências de caracteres com um tamanho que não é fixo. Nas descrições destes parâmetros, um  $n$  minúsculo representa uma constante numérica. Ao codificar a declaração para esse parâmetro, substitua o  $n$  pelo valor numérico necessário.

## **MQBACK (Mudanças de retorno) em IBM i**

A chamada MQBACK indica para o gerenciador de filas que todas as mensagens obtêm e colocam que ocorreram desde o último ponto de sincronização devem ser restauradas. Mensagens colocadas como parte de uma unidade de trabalho são excluídas; mensagens recuperadas como parte de uma unidade de trabalho são restabelecidas na fila.

- Essa chamada é suportada nos seguintes ambientes:

-  AIX
-  IBM i
-  Windows

- [“Sintaxe” na página 1296](#)
- [“Observações de Uso” na página 1297](#)
- [“Parâmetros” na página 1298](#)
- [“Declaração de RPG” na página 1299](#)

#### **Sintaxe**

MQBACK (*Hconn*, *CompCode*, *Reason*)

## Observações de Uso

Considere estas notas de uso ao usar o MQBACK

1. Essa chamada pode ser usada apenas quando o próprio gerenciador de filas coordena a unidade de trabalho. Esta é uma unidade de trabalho local, em que as alterações afetam apenas recursos IBM MQ.
2. Em ambientes nos quais o gerenciador de filas não coordena a unidade de trabalho, a chamada de retorno apropriada deve ser usada em vez de MQBACK. O ambiente também pode suportar um retorno implícito causado pela finalização anormal do aplicativo.
  - No IBM i, essa chamada pode ser usada para unidades locais de trabalho coordenadas pelo gerenciador de fila.. Isso significa que uma definição de confirmação não deve existir no nível da tarefa, ou seja, o comando STRCMTCTL com o parâmetro **CMTSCOPE (\*JOB)** não deve ter sido emitido para a tarefa
3. Se um aplicativo terminar com mudanças não confirmados em uma unidade de trabalho, a disposição dessas mudanças dependerá se o aplicativo será encerrado normalmente ou de forma anormal. Consulte as notas de uso em [“MQDISC \(Desconectar o gerenciador de filas\) no IBM i”](#) na página 1336 para obter detalhes adicionais
4. Quando um aplicativo coloca ou obtém mensagens em grupos ou segmentos de mensagens lógicas, o gerenciador de filas retém informações relacionadas ao grupo de mensagens e mensagens lógicas para as últimas chamadas MQPUT e MQGET bem-sucedidas. Essas informações estão associadas ao identificador de filas e incluem itens como:
  - Os valores dos campos *MDGID*, *MDSEQ*, *MDOFF* e *MDMFL* em MQMD..
  - Indica se a mensagem faz parte de uma unidade de trabalho
  - Para a chamada MQPUT: se a mensagem é persistente ou não persistente.

O gerenciador de filas mantém *três* conjuntos de informações de grupo e segmento, um conjunto para cada um dos seguintes:

- A última chamada MQPUT bem-sucedida (isso pode fazer parte de uma unidade de trabalho)
- A última chamada MQGET bem-sucedida que removeu uma mensagem da fila (isso pode fazer parte de uma unidade de trabalho).
- A última chamada MQGET bem-sucedida que navegou em uma mensagem na fila (isso não pode fazer parte de uma unidade de trabalho)

Se o aplicativo colocar ou receber as mensagens como parte de uma unidade de trabalho e o aplicativo decidir restaurar a unidade de trabalho, as informações do grupo e do segmento serão restauradas para o valor que ele tinha anteriormente:

- As informações associadas à chamada MQPUT são restauradas para o valor que ela tinha antes da primeira chamada MQPUT bem-sucedida para essa manipulação de fila na unidade de trabalho atual..
- As informações associadas à chamada MQGET são restauradas para o valor que ela tinha antes da primeira chamada MQGET bem-sucedida para essa manipulação de fila na unidade de trabalho atual...

As filas que foram atualizadas pelo aplicativo após a unidade de trabalho ter sido iniciada, mas fora do escopo da unidade de trabalho, não terão suas informações de grupo e de segmento restauradas se a unidade de trabalho for restaurada.

A restauração das informações do grupo e do segmento para seu valor anterior quando uma unidade de trabalho é restaurada permite que o aplicativo difunda um grupo de mensagens grandes ou uma mensagem lógica grande que consiste em muitos segmentos em várias unidades de trabalho e reinicie no ponto correto no grupo de mensagens ou mensagem lógica se uma das unidades de trabalho falhar. Usar várias unidades de trabalho pode ser vantajoso se o gerenciador de fila local tiver apenas armazenamento de fila limitado. No entanto, o aplicativo deve manter informações suficientes para poder reiniciar a colocação ou obtenção de mensagens no ponto correto se ocorrer uma falha do sistema. Para obter detalhes sobre como reiniciar no ponto correto depois de uma falha do sistema,

consulte a opção PMLOGO descrita em “MQPMO (opções Put-message) em IBM i” na página 1215 e a opção GMLOGO descrita em “MQGMO (opções Get-message) em IBM i” na página 1113

As notas de uso restantes se aplicam apenas quando o gerenciador de filas coordena as unidades de trabalho:

1. Uma unidade de trabalho tem o mesmo escopo que uma manipulação de conexões.. Isso significa que todas as chamadas do IBM MQ que afetam uma unidade de trabalho específica devem ser executadas usando o mesmo identificador de conexões Chamadas emitidas usando um identificador de conexão diferente (por exemplo, chamadas emitidas por outro aplicativo) afetam uma unidade de trabalho diferente. Consulte o parâmetro **HCONN** descrito em “MQCONN (Conectar gerenciador de filas) no IBM i” na página 1323 para obter informações sobre o escopo de manipulações de conexão
2. Apenas as mensagens que foram colocadas ou recuperadas como parte da unidade de trabalho atual são afetada por esta chamada
3. Um aplicativo de longa execução que emite chamadas MQGET ou MQPUT ou MQPUT1 dentro de uma unidade de trabalho, mas que nunca emite uma confirmação ou chamada de restauração, pode fazer com que as filas sejam preenchidas com mensagens que não estão disponíveis para outros aplicativos. Para proteger contra essa possibilidade, o administrador deve configurar o atributo do gerenciador de filas **MaxUncommittedMsgs** para um valor que seja baixo o suficiente para evitar que aplicativos runaway preencham as filas, mas alto o suficiente para permitir que os aplicativos do sistema de mensagens esperados funcionem corretamente.

## Parâmetros

A chamada MQBACK possui os parâmetros a seguir:

### **HCONN (número inteiro assinado de 10 dígitos)-entrada**

Manipulação de conexões.

Esta manipulação representa a conexão ao gerenciador de filas. O valor de *HCONN* foi retornado por uma chamada anterior MQCONN ou MQCONNX.

### **CMPCOD (número inteiro assinado de 10 dígitos)-saída**

Código de conclusão.

É um dos seguintes:

#### **CCOK**

Indica conclusão bem-sucedida.

#### **CCFAIL**

A chamada falhou.

### **REASON (número inteiro assinado de 10 dígitos)-saída**

Código de razão qualificando *COMCOD*.

Se *COMCOD* for CCOK:

#### **RCNONE**

(0, X'000') Nenhuma razão a relatar.

Se *COMCOD* for CCFAIL:

#### **RC2219**

(2219, X'8AB') chamada MQI reinserida antes da chamada anterior ser concluída.

#### **RC2009**

(2009, X'7D9') Conexão ao gerenciador de filas perdida.

#### **RC2018**

(2018, X'7E2') Manipulação de conexões não válida.

#### **RC2101**

(2101, X'835') Objeto danificado.

**RC2123**

(2123, X'84B') O resultado da operação de confirmação ou de retorno é combinado.

**RC2162**

(2162, X'872') Gerenciador de filas sendo encerrado.

**RC2102**

(2102, X'836') Recursos insuficientes do sistema disponíveis.

**RC2071**

(2071, X'817') Armazenamento insuficiente disponível.

**RC2195**

(2195, X'893') Ocorreu um erro inesperado.

**Declaração de RPG**

```

C*..1.....2.....3.....4.....5.....6.....7..
C          CALLP      MQBACK(HCONN : COMCOD : REASON)

```

A definição de protótipo para a chamada é:

```

D*..1.....2.....3.....4.....5.....6.....7..
DMQBACK      PR          EXTPROC('MQBACK')
D* Connection handle
D HCONN          10I 0 VALUE
D* Completion code
D COMCOD          10I 0
D* Reason code qualifying COMCOD
D REASON          10I 0

```

**IBM i MQBEGIN (Iniciar unidade de trabalho) em IBM i**

A chamada MQBEGIN inicia uma unidade de trabalho coordenada pelo gerenciador de filas e que pode envolver gerenciadores de recursos externos.

- Essa chamada é suportada nos seguintes ambientes:

-  AIX
-  IBM i
-  Windows

- [“Sintaxe” na página 1299](#)
- [“Observações de Uso” na página 1299](#)
- [“Parâmetros” na página 1301](#)
- [“Declaração de RPG” na página 1302](#)

**Sintaxe**

MQBEGIN (*HCONN*, *BEGOP*, *CMPCOD*, *REASON*)

**Observações de Uso**

1. A chamada MQBEGIN pode ser usada para iniciar uma unidade de trabalho que é coordenada pelo gerenciador de filas e que pode envolver mudanças em recursos de propriedade de outros gerenciadores de recursos O gerenciador de filas suporta três tipos de unidade de trabalho:

### **Gerenciador de filas-unidade de trabalho local coordenada**

Esta é uma unidade de trabalho na qual o gerenciador de fila é o único gerenciador de recursos participante e, portanto, o gerenciador de fila age como o coordenador da unidade de trabalho.

- Para iniciar esse tipo de unidade de trabalho, a opção PMSYP ou GMSYP deve ser especificada na primeira chamada MQPUT, MQPUT1 ou MQGET na unidade de trabalho.

Não é necessário que o aplicativo emita a chamada MQBEGIN para iniciar a unidade de trabalho, mas se MQBEGIN for usado, a chamada será concluída com CCWARN e o código de razão RC2121.

- Para confirmar ou voltar esse tipo de unidade de trabalho, a chamada de MQCMIT ou MQBACK deve ser usada

### **Gerenciador de filas-unidade de trabalho global coordenada**

Esta é uma unidade de trabalho na qual o gerenciador de filas age como o coordenador da unidade de trabalho, para IBM MQ recursos e para recursos pertencentes a outros gerenciadores de recursos. Esses gerenciadores de recursos cooperam com o gerenciador de filas para assegurar que todas as mudanças nos recursos na unidade de trabalho sejam confirmados ou restaurados juntos

- Para iniciar esse tipo de unidade de trabalho, a chamada MQBEGIN deve ser usada..
- Para confirmar ou voltar esse tipo de unidade de trabalho, as chamadas MQCMIT e MQBACK devem ser usadas

### **Unidade de trabalho global coordenada externamente**

Esta é uma unidade de trabalho na qual o gerenciador de fila é um participante, mas o gerenciador de filas não age como o coordenador da unidade de trabalho. Em vez disso, há um coordenador de unidade de trabalho externo com o qual o gerenciador de filas coopera

- Para iniciar este tipo de unidade de trabalho, deve ser utilizada a chamada relevante fornecida pelo coordenador externo da unidade de trabalho.

Se a chamada MQBEGIN for usada para tentar iniciar a unidade de trabalho, a chamada falhará com o código de razão RC2012..

- Para confirmar ou voltar este tipo de unidade de trabalho, as chamadas de confirmação e de retorno fornecidas pelo coordenador externo da unidade de trabalho devem ser usadas.

Se a chamada MQCMIT ou MQBACK for usada para tentar confirmar ou voltar para a unidade de trabalho, a chamada falhará com o código de razão RC2012

2. Se o aplicativo terminar com mudanças não confirmados em uma unidade de trabalho, a disposição dessas mudanças dependerá se o aplicativo será encerrado normalmente ou anormalmente, ou não. Consulte as notas de uso em [“MQDISC \(Desconectar o gerenciador de filas\) no IBM i”](#) na página 1336 para obter detalhes adicionais
3. Um aplicativo pode participar de apenas uma unidade de trabalho por vez.. A chamada MQBEGIN falha com o código de razão RC2128 se já houver uma unidade de trabalho existente para o aplicativo, independentemente de qual tipo de unidade de trabalho ele é.
4. A chamada MQBEGIN não é válida em um ambiente do cliente do IBM MQ. Uma tentativa de usar a chamada falha com o código de razão RC2012
5. Quando o gerenciador de filas estiver agindo como o coordenador da unidade de trabalho para as unidades globais de trabalho, os gerenciadores de recurso que podem participar da unidade de trabalho serão definidos no arquivo de configuração do gerenciador de filas
6. No IBM i, os três tipos de unidade de trabalho são suportados da seguinte forma:
  - **Unidades locais de trabalho coordenadas pelo gerenciador de filas** podem ser usadas apenas quando uma definição de confirmação não existir no nível da tarefa, ou seja o comando STRCMTCTL com o parâmetro **CMTSCOPE(\*JOB)** não deve ter sido emitido para a tarefa
  - **Unidades globais de trabalho coordenadas pelo gerenciador de filas** não são suportadas.
  - **Unidades globais de trabalho coordenadas externamente** podem ser usadas somente quando uma definição de confirmação existir no nível da tarefa, ou seja, o comando STRCMTCTL com



o parâmetro **CMTSCOPE (\*JOB)** deve ter sido emitido para a tarefa. Se isso tiver sido feito, as operações IBM i COMMIT e ROLLBACK se aplicam a recursos IBM MQ, bem como a recursos pertencentes a outros gerenciadores de recursos participantes.

## Parâmetros

A chamada MQBEGIN possui os seguintes parâmetros:

### **HCONN (número inteiro assinado de 10 dígitos)-entrada**

Manipulação de conexões.

Esta manipulação representa a conexão ao gerenciador de filas. O valor de *HCONN* foi retornado por uma chamada anterior MQCONN ou MQCONNX.

### **BEGOP (MQBO)-entrada/saída**

As opções que controlam a ação de MQBEGIN

Consulte “MQBO (Iniciar opções) em IBM i” na página 1053 para obter detalhes.

Se nenhuma opção for necessária, os programas gravados no assembler C ou S/390 poderão especificar um endereço de parâmetro nulo, em vez de especificar o endereço de uma estrutura MQBO.

### **CMPCOD (número inteiro assinado de 10 dígitos)-saída**

Código de conclusão.

É um dos seguintes:

#### **CCOK**

Indica conclusão bem-sucedida.

#### **CCWARN**

Aviso (conclusão parcial).

#### **CCFAIL**

A chamada falhou.

### **REASON (número inteiro assinado de 10 dígitos)-saída**

Código de razão qualificando *CMPCOD*.

Se *CMPCOD* for CCOK:

#### **RCNONE**

(0, X'000') Nenhuma razão a relatar.

Se *CMPCOD* for CCWARN:

#### **RC2121**

(2121, X'849 ') Nenhum gerenciador de recursos participante registrado.

#### **RC2122**

(2122, X'84A') O gerenciador de recursos participante não está disponível.

Se *CMPCOD* for CCFAIL:

#### **RC2134**

(2134, X'856 ') Estrutura de opções de início inválida.

#### **RC2219**

(2219, X'8AB') chamada MQI reinserida antes da chamada anterior ser concluída.

#### **RC2009**

(2009, X'7D9') Conexão ao gerenciador de filas perdida.

#### **RC2012**

(2012, X'7DC') Chamada não válida no ambiente.

**RC2018**

(2018, X'7E2') Manipulação de conexões não válida.

**RC2046**

(2046, X'7FE') Opções não válidas ou não consistentes.

**RC2162**

(2162, X'872') Gerenciador de filas sendo encerrado.

**RC2102**

(2102, X'836') Recursos insuficientes do sistema disponíveis.

**RC2071**

(2071, X'817') Armazenamento insuficiente disponível.

**RC2195**

(2195, X'893') Ocorreu um erro inesperado.

**RC2128**

(2128, X'850 ') Unidade de trabalho já iniciada.

**Declaração de RPG**

```

C*.1.....2.....3.....4.....5.....6.....7..
C          CALLP      MQBEGIN(HCONN : BEGOP : CMPCOD :
C                                REASON)

```

A definição de protótipo para a chamada é:

```

D*.1.....2.....3.....4.....5.....6.....7..
DMQBEGIN      PR          EXTPROC('MQBEGIN')
D* Connection handle
D HCONN              10I 0 VALUE
D* Options that control the action of MQBEGIN
D BEGOP              12A
D* Completion code
D CMPCOD              10I 0
D* Reason code qualifying CMPCOD
D REASON              10I 0

```

## MQBUFMH (Converter buffer em identificador de mensagens) no IBM i

A chamada de função MQBUFMH converte um buffer em uma manipulação de mensagens e é o inverso da chamada MQMHBUF.

Essa chamada usa um descritor de mensagem e propriedades MQRFH2 no buffer e os disponibiliza por meio de um identificador de mensagem.. As propriedades MQRFH2 nos dados da mensagem são, opcionalmente, removidas. Os campos *Encoding*, *CodedCharSetIde Format* do descritor de mensagens são atualizados, se necessário, para descrever corretamente o conteúdo do buffer após a remoção das propriedades.

- [“Sintaxe” na página 1302](#)
- [“Observações de Uso” na página 1303](#)
- [“Parâmetros” na página 1303](#)
- [“Declaração de RPG” na página 1304](#)

**Sintaxe**

MQBUFMH (*Hconn*, *Hmsg*, *BufMsgHOpts*, *MsgDesc*, *Buffer*, *BufferLength*, *DataLength*, *CompCode*, *Reason*)

## Observações de Uso

As chamadas MQBUFMH não podem ser interceptadas por saídas de API-um buffer é convertido em um identificador de mensagem no espaço do aplicativo; a chamada não atinge o gerenciador de filas

## Parâmetros

A chamada MQBUFMH possui os parâmetros a seguir:

### HCONN (número inteiro assinado de 10 dígitos)-entrada

Esta manipulação representa a conexão ao gerenciador de filas. O valor de *HCONN* deve corresponder à manipulação de conexões que foi usada para criar a manipulação de mensagens especificada no parâmetro **Hmsg** ..

Se o identificador de mensagens tiver sido criado usando HCUNAS, uma conexão válida deverá ser estabelecida no encadeamento convertendo um buffer em um identificador de mensagens Se uma conexão válida não for estabelecida a chamada falhará com RC2009.

### HMSG (número inteiro assinado de 20 dígitos)-entrada

Esse identificador é o identificador de mensagem para o qual um buffer é necessário O valor foi retornado por uma chamada MQCRTMH anterior.

### BMHOPT (MQBMHO)-entrada

A estrutura MQBMHO permite que os aplicativos especifiquem opções que controlam como as manipulações de mensagens são produzidas a partir de buffers

Consulte [“MQBMHO \(Buffer para opções de manipulação de mensagens\) no IBM i” na página 1052](#) para obter detalhes.

### MSGDSC (MQMD)-entrada/saída

A estrutura *MSGDSC* contém as propriedades do descritor de mensagem e descreve o conteúdo da área de buffer..

Na saída da chamada, as propriedades são removidas opcionalmente da área de buffer e, nesse caso, o descritor de mensagem é atualizado para descrever corretamente a área de buffer

Os dados nessa estrutura devem estar no conjunto de caracteres e na codificação do aplicativo

### BUFLEN (número inteiro assinado de 10 dígitos)-entrada

*BUFLEN* é o comprimento da área Buffer, em bytes.

Um *BUFLEN* de zero bytes é válido e indica que a área do buffer não contém dados.

### BUFFER (cadeia de bits de 1 byte x BUFLEN)-entrada/saída

*BUFFER* define a área contendo o buffer de mensagem. Para a maioria dos dados, você deve alinhar o buffer em um limite de 4 bytes.

Se o *BUFFER* contiver dados de caractere ou numéricos, configure os campos *CodedCharSetId* e *Encoding* no parâmetro **MSGDSC** para os valores apropriados para os dados; isso permite que os dados sejam convertidos, se necessário,.

Se as propriedades forem localizadas no buffer de mensagem, elas serão removidas opcionalmente; posteriormente, elas se tornarão disponíveis a partir do identificador de mensagem no retorno da chamada

Na linguagem de programação C, o parâmetro é declarado como um ponteiro para vazio, o que significa que o endereço de qualquer tipo de dados pode ser especificado como o parâmetro.

Se o parâmetro **BUFLEN** for zero, *BUFFER* não será referido. Nesse caso, o endereço de parâmetro transmitido por programas gravados em C ou assembler System/390 pode ser nulo.

## DATLEN (número inteiro assinado de 10 dígitos)-saída

DATLEN é o comprimento, em bytes, do buffer que pode ter as propriedades removidas..

## CMPCOD (número inteiro assinado de 10 dígitos)-saída

### CCOK

Indica conclusão bem-sucedida.

### CCFAIL

A chamada falhou.

## REASON (número inteiro assinado de 10 dígitos)-saída

O código de razão qualificando CMPCOD.

Se CMPCOD for CCOK:

### RCNONE

(0, X'000') Nenhuma razão a relatar.

Se CMPCOD for CCFAIL:

### RC2204

(2204, X'089C') Adaptador não disponível.

### RC2130

(2130, X'852') Não foi possível carregar o módulo de serviço do adaptador.

### RC2157

(2157, X'86D') ASIDs de início e primárias são diferentes.

### RC2489

(2489, X'09B9') Buffer para estrutura de opções de manipulação de mensagens não é válido.

### RC2004

(2004, X'07D4') Parâmetro de buffer inválido.

### RC2005

(2005, X'07D5') Parâmetro de comprimento do buffer inválido.

### RC2219

(2219, X'08AB') chamada MQI inserida antes da chamada anterior ser concluída.

### RC2009

(2009, X'07D9') Conexão com o gerenciador de filas perdida.

### RC2460

(2460, X'099C') Identificador de mensagem inválido.

### RC2026

(2026, X'07EA') Descritor de mensagens inválido.

### RC2499

(2499, X'09C3') Identificador de mensagem já em uso.

### RC2046

(2046, X'07FE') Opções não válidas ou não consistentes.

### RC2334

(2334, X'091E') MQRFH2 estrutura não válida.

### RC2421

(2421, X'0975 ') Uma pasta MQRFH2 contendo propriedades não pôde ser analisada.

### RC2195

(2195, X'893') Ocorreu um erro inesperado.

## Declaração de RPG

```
C*..1.....2.....3.....4.....5.....6.....7..
C          CALLP      MQBUFMH(HCONN : HMSG : BMHOPT :
```

```
MSGDSC : BUFLen : BUFFER :  
DATLEN : CMPCOD : REASON)
```

A definição de protótipo para a chamada é:

```
DMQBUFMH          PR          EXTPROC('MQBUFMH')  
D* Connection handle  
D HCONN           10I 0  
D* Message handle  
D HMSG           10I 0  
D* Options that control the action of MQBUFMH  
D BMHOPT          12A VALUE  
D* Message descriptor  
D MSGDSC          364A  
D* Length in bytes of the Buffer area  
D BUFLen          10I 0  
D* Area to contain the message buffer  
D BUFFER          * VALUE  
D* Length of the output buffer  
D DATLEN          10I 0  
D* Completion code  
D CMPCOD          10I 0  
D* Reason code qualifying CompCode  
D REASON          10I 0
```

IBM i

## MQCB (Gerenciar retorno de chamada) no IBM i

A chamada MQCB registra novamente um retorno de chamada para o identificador de objeto especificado e controla a ativação e as mudanças para o retorno de chamada

Um retorno de chamada é um pedaço de código (especificado como o nome de uma função que pode ser vinculada dinamicamente ou como um ponteiro de função) que é chamado por IBM MQ quando determinados eventos ocorrem.

Para usar MQCB e MQCTL em um cliente V7, você deve estar conectado a um servidor V7 e o parâmetro **SHARECNV** do canal deve ter um valor diferente de zero.

Para obter informações sobre unidades globais de trabalho, consulte: [Unidades globais de trabalho](#).

Os tipos de retorno de chamada que podem ser definidos são:

### Consumidor de mensagens

Uma função de retorno de chamada do consumidor de mensagens é chamada quando uma mensagem, atendendo aos critérios de seleção especificados, está disponível em uma manipulação de objetos...

Apenas uma função de retorno de chamada pode ser registrada em cada identificador de objeto. Se uma única fila tiver que ser lida com diversos critérios de seleção, a fila deverá ser aberta várias vezes e uma função do consumidor registrada em cada identificador.

### Manipulador de eventos

O manipulador de eventos é chamado para as condições que afetam todo o ambiente de retorno de chamada.

A função é chamado quando ocorre uma condição de evento, por exemplo, um gerenciador de filas ou conexão parando ou quiesce.

A função não é chamado para condições que são específicas para um único consumidor de mensagens, por exemplo, RC2016; é chamado, no entanto, se uma função de retorno de chamada não terminar normalmente.

- [“Sintaxe” na página 1306](#)
- [“Notas de uso para MQCB” na página 1306](#)
- [“Parâmetros para MQCB” na página 1307](#)
- [“Declaração de RPG” na página 1313](#)

## Sintaxe

MQCB (HCONN, OPERATN, HOBJ, CBDSC, MSGDSC, GMO, CMPCOD, REASON)

## Notas de uso para MQCB

1. MQCB é usado para definir a ação a ser chamada para cada mensagem, correspondendo aos critérios especificados, disponível na fila. Quando a ação é processada, a mensagem é removida da fila e transmitida ao consumidor de mensagens definido ou um token de mensagem é fornecido, que é usado para recuperar a mensagem.
2. MQCB pode ser usado para definir rotinas de retorno de chamada antes de iniciar o consumo com MQCTL ou pode ser usado a partir de uma rotina de retorno de chamada.
3. Para usar MQCB de fora de uma rotina de retorno de chamada, você deve primeiro suspender o consumo de mensagens usando MQCTL e continuar o consumo posteriormente.

## Sequência de retorno de chamada do consumidor de mensagens

É possível configurar um consumidor para chamar o retorno de chamada em pontos-chave durante o ciclo de vida do consumidor. Por exemplo:

- quando o consumidor é registrado pela primeira vez,
- quando a conexão é iniciada,
- quando a conexão for interrompida e
- quando o consumidor tem o registro removido, seja explicitamente ou implicitamente por um MQCLOSE

Verb	Significado
MQCTL (START)	Chamada MQCTL usando a Operação CTLSR
MQCTL (STOP)	Chamada MQCTL usando a Operação CTLSP
MQCTL (WAIT)	Chamada MQCTL usando a Operação CTLSW

Permite ao consumidor manter o estado associado ao consumidor. Quando um retorno de chamada é solicitado por um aplicativo, as regras para chamada do consumidor são as seguintes:

### REGISTRE-SE

É sempre o primeiro tipo de chamada do retorno de chamada

É sempre chamado no mesmo encadeamento da chamada MQCB (CBREG).

### START

É sempre chamado sincronicamente com o verbo MQCTL (START).

- Todos os retornos de chamada START são concluídos antes do retorno do verbo MQCTL (START).

Está no mesmo encadeamento que a entrega de mensagem, se CTLTHR for solicitado

A chamada com início não será garantida se, por exemplo, um retorno de chamada anterior emitir MQCTL (STOP) durante o MQCTL (START)

### PARAR

Nenhuma mensagem ou evento adicional será entregue após essa chamada até que a conexão seja reiniciada

Um STOP é garantido se o aplicativo foi chamado anteriormente para START ou uma mensagem ou um evento.

### DEREGISTER

É sempre o último tipo de chamada do retorno de chamada

Assegure-se de que seu aplicativo execute inicialização e limpeza baseadas em encadeamento nos retornos de chamada START e STOP. É possível executar inicialização e limpeza não baseadas em encadeamento com retornos de chamadas REGISTER e REMOVE ister.

Não faça nenhuma suposição sobre a vida e a disponibilidade da linha além do que é declarado. Por exemplo, não confie em um encadeamento que permanece ativo além da última chamada para REMOVER o Registro. Da mesma forma, quando você tiver escolhido não usar o CTLTHR, não assuma que o encadeamento existe sempre que a conexão for iniciada.

Se o seu aplicativo tiver requisitos específicos para características de encadeamento, ele sempre poderá criar um encadeamento de acordo, em seguida, usar MQCTL (WAIT). Esta etapa *doa* o encadeamento para IBM MQ para entrega de mensagem assíncrona.

### **Uso da conexão do consumidor da mensagem**

Normalmente, quando um aplicativo emite outra chamada MQI enquanto uma está pendente, a chamada falha com código de razão RC2219.

Há casos especiais, no entanto, quando o aplicativo deve emitir uma chamada MQI adicional antes da chamada anterior ser concluída. Por exemplo, o consumidor pode ser chamado durante uma chamada MQCB com a CBRE.

Em tal instância, quando como resultado do aplicativo emitir um verbo MQCB ou MQCTL, o aplicativo é chamado de volta, o aplicativo tem permissão para emitir uma chamada MQI adicional. Essa instância significa que é possível emitir, por exemplo, uma chamada MQOPEN na função do consumidor quando chamada com um tipo CBCCALLT de CBCTRC. Qualquer chamada MQI, exceto MQDISC, é permitida.

### **Parâmetros para MQCB**

A chamada MQCB possui os parâmetros a seguir:

#### **HCONN (número inteiro assinado de 10 dígitos)-entrada**

Gerenciar função de retorno de chamada-parâmetro HCONN.

Esta manipulação representa a conexão ao gerenciador de filas. O valor de *HCONN* foi retornado por uma chamada anterior MQCONN ou MQCONNX.

#### **OPERATN (número inteiro assinado de 10 dígitos)-entrada**

Gerenciar função de retorno de chamada-parâmetro OPERATN.

A operação sendo processada no retorno de chamada definido para o identificador de objeto especificado. Você deve especificar uma das opções a seguir; se mais de uma opção for necessária, os valores poderão ser incluídos (não inclua a mesma constante mais de uma vez) ou combinados usando a operação OR bit a bit (se a linguagem de programação suportar operações bit a bit).

As combinações inválidas são observadas; todas as outras combinações são válidas.

#### **CBREG**

Defina a função de retorno de chamada para a manipulação de objetos especificada.. Esta operação define a função a ser chamada e os critérios de seleção a serem usados.

Se uma função de retorno de chamada já tiver sido definida para a manipulação de objetos, a definição será substituída Se um erro for detectado ao substituir o retorno de chamada, a função será removidas do registro..

Se um retorno de chamada for registrado na mesma função de retorno de chamada na qual ele foi removido anteriormente, isso será tratado como uma operação de substituição; quaisquer chamadas iniciais ou finais não serão chamadas.

É possível usar CBREG com CTLSU ou CTLRE.

#### **CBUNR**

Pare o consumo de mensagens para a manipulação de objetos e remove a manipulação daqueles elegíveis para um retorno de chamada

Um retorno de chamada será removido automaticamente se o identificador associado for fechado.

Se CBUNR for chamado de dentro de um consumidor e o retorno de chamada tiver uma chamada de parada definida, ele será chamado no retorno do consumidor.

Se essa operação for emitida em um *Hobj* sem consumidor registrado, a chamada retornará com RC2448.

### **CTLSU**

Suspende o consumo de mensagens para a manipulação de objetos

Se essa operação for aplicada a um manipulador de eventos, o manipulador de eventos não obterá eventos enquanto estiver suspenso e quaisquer eventos perdidos enquanto estiver no estado suspenso não serão fornecidos para a operação quando ela for retomada.

Enquanto suspensa, a função do consumidor continua a obter os retornos de chamada de tipo de controle.

### **CTLRE**

Retome o consumo de mensagens para a manipulação de objetos

Se essa operação for aplicada a um manipulador de eventos, o manipulador de eventos não obterá eventos enquanto estiver suspenso e quaisquer eventos perdidos enquanto estiver no estado suspenso não serão fornecidos para a operação quando ela for retomada.

### **CBDSC (MQCBD)-entrada**

Gerenciar função de retorno de chamada-parâmetro CBDSC.

Essa é uma estrutura que identifica a função de retorno de chamada que está sendo registrada pelo aplicativo e as opções usadas ao registrá-la,

Consulte “[MQCBD-Descriptor de retorno de chamada](#)” na página 295 para obter detalhes da estrutura..

O descriptor de retorno de chamada é necessário apenas para a opção CBREG; se o descriptor não for necessário, o endereço do parâmetro transmitido pode ser nulo.

### **HOBJ (número inteiro assinado de 10 dígitos)-entrada**

Gerenciar função de retorno de chamada-parâmetro HOBJ.

Essa manipulação representa o acesso que foi estabelecido para o objeto do qual uma mensagem deve ser consumida. Esta é uma manipulação que foi retornada de uma chamada anterior [MQOPEN](#) ou [MQSUB](#) (no parâmetro **HOBJ**).

*HOBJ* não é necessário ao definir uma rotina do manipulador de eventos (CBTEH) e deve ser especificado como HONONE.

Se este *Hobj* tiver sido retornado de uma chamada [MQOPEN](#), a fila deverá ter sido aberta com uma ou mais das seguintes opções:

- OOINPS
- OOINPX
- OOINPQ
- OOBW

### **MSGDSC (MQMD)-entrada**

Gerenciar a função de retorno de chamada do parâmetro -MSGDSC.

Essa estrutura descreve os atributos da mensagem requerida e os atributos da mensagem recuperada.

O parâmetro **MsgDesc** define os atributos das mensagens requeridas pelo consumidor e a versão do MQMD a ser transmitida para o consumidor de mensagem

O *MsgId*, *CorrelId*, *GroupId*, *MsgSeqNumbere Offset* no MQMD são usados para seleção de mensagens dependendo das opções especificadas no parâmetro **GetMsgOpts**.



O *Encoding* e o *CodedCharSetId* serão usados para conversão de mensagens se você especificar a opção GMCONV

Consulte [MQMD](#) para obter detalhes..

*MsgDesc* é usado apenas para CBREG e, se você precisar de valores diferentes do padrão para quaisquer campos, O *MsgDesc* não é usado para um manipulador de eventos

Se o descritor não for necessário, o endereço do parâmetro transmitido poderá ser nulo.

Observe que, se vários consumidores forem registrados na mesma fila com seletores de sobreposição, o consumidor escolhido para cada mensagem será indefinido.

## **GMO (MQGMO)-entrada**

Gerenciar função de retorno de chamada-parâmetro GMO.

Opções que controlam como o consumidor de mensagens obtém as mensagens

Todas as opções têm o significado conforme descrito em [“MQGMO \(opções Get-message\) em IBM i”](#) na página 1113, quando usado em uma chamada MQGET, exceto:

### **GMSSIG**

Esta opção não é permitida

### **GMBRWF, GMBRWN, GMMBH, GMMBC**

A ordem das mensagens entregues a um consumidor de navegação é ditada pelas combinações dessas opções.. Combinações significativas são:

#### **GMBRWF**

A primeira mensagem na fila é entregue repetidamente para o consumidor Isso é útil quando o consumidor consome destrutivamente a mensagem no retorno de chamada. Use esta opção com cuidado..

#### **GMBRWN**

O consumidor recebe cada mensagem na fila, da posição atual do cursor até o final da fila ser alcançado.

#### **GMBRWF + GMBRWN**

O cursor é reconfigurado para o início da fila O consumidor recebe então cada mensagem até que o cursor atinja o final da fila.

#### **GMBRWF + GMMBH ou GMMBC**

A partir do início da fila, o consumidor recebe a primeira mensagem não marcada na fila, que é então marcada para esse consumidor. Essa combinação assegura que o consumidor possa receber novas mensagens incluídas atrás do ponto do cursor atual..

#### **GMBRWN + GMMBH ou GMMBC**

Iniciando na posição do cursor, o consumidor recebe a próxima mensagem não marcada na fila, que é então marcada para esse consumidor. Utilize esta combinação com cuidado porque as mensagens podem ser adicionadas à fila atrás da posição atual do cursor.

#### **GMBRWF + GMBRWN + GMMBH ou GMMBC**

Essa combinação não será permitida, se usada, a chamada retornará RC2046

## **GMNWT, GMWT e GMWI**

Essas opções controlam como o consumidor é chamado.

### **GMNWT**

O consumidor nunca é chamado com RC2033.. O consumidor é chamado apenas para mensagens e eventos

### **GMWT com um GMWI zero**

O código RC2033 é transmitido apenas para o consumidor quando não há mensagens e

- o consumidor foi iniciado
- o consumidor foi entregue pelo menos uma mensagem desde o último código de razão sem mensagens.

Isso impede o consumidor de pesquisar em um loop ocupado quando um intervalo de espera zero é especificado.

#### **GMWT e um GMWI positivo**

O usuário é chamado após o intervalo de espera especificado com o código de razão RC2033. Essa chamada é feita independentemente se alguma mensagem foi entregue ao consumidor.. Isso permite que o usuário execute o processamento de pulsação ou de tipo de lote

#### **GMWT e GMWI do WIULIM**

Especifica uma espera infinita antes de retornar RC2033. O consumidor nunca é chamado com RC2033..

*GMO* é usado apenas para CBREG e, se você precisar de valores diferentes do padrão para quaisquer campos, O *GMO* não é usado para um manipulador de eventos

Se as opções não forem necessárias, o endereço do parâmetro transmitido poderá ser nulo..

Se uma manipulação de propriedades de mensagem for fornecida na estrutura MQGMO, uma cópia será fornecida na estrutura MQGMO que é transmitida para o retorno de chamada do consumidor No retorno da chamada MQCB, o aplicativo pode excluir o identificador de propriedades da mensagem

### **CMPCOD (número inteiro assinado de 10 dígitos)-saída**

Gerenciar função de retorno de chamada-parâmetro CMPCOD.

O código de conclusão; é um dos seguintes:

#### **CCOK**

Indica conclusão bem-sucedida.

#### **CCWARN**

Aviso (conclusão parcial).

#### **CCFAIL**

A chamada falhou.

### **REASON (número inteiro assinado de 10 dígitos)-saída**

Gerenciar função de retorno de chamada-parâmetro REASON.

Os seguintes códigos de razão são os códigos que o gerenciador de fila pode retornar para o parâmetro **REASON** .

Se *CMPCOD* for CCOK:

#### **RCNONE**

(0, X'000') Nenhuma razão a relatar.

Se *CompCode* for CCFAIL:

#### **RC2204**

(2204, X'89C') Adaptador não disponível.

#### **RC2133**

(2133, X'855') Impossível carregar módulos de serviços de conversão de dados.

#### **RC2130**

(2130, X'852') Não foi possível carregar o módulo de serviço do adaptador.

#### **RC2374**

(2374, X'946') Falha na saída de API.

#### **RC2183**

(2183, X'887') Não foi possível carregar a saída de API.

#### **RC2157**

(2157, X'86D') ASIDs de início e primárias são diferentes.

#### **RC2005**

(2005, X'7D5') Parâmetro de comprimento de buffer inválido.

**RC2219**

(2219, X'8AB') Chamada MQI inserida antes da chamada anterior concluída.

**RC2487**

(2487, X'9B7') Campo de tipo de retorno de chamada incorreto

**RC2448**

(2448, X' 990 ') Não é possível remover o registro, suspender ou continuar, porque não há retorno de chamada registrado

**RC2486**

(2486, X'9B6') *CallbackFunction* ou *CallbackName* deve ser especificado, mas não ambos.

**RC2483**

(2483, X'9B3') Campo de tipo de retorno de chamada incorreto

**RC2484**

(2484, X'9B4') Campo de opções MQCBD incorreto.

**RC2140**

(2140, X'85C') Pedido de espera rejeitado por CICS.

**RC2009**

(2009, X'7D9') Conexão ao gerenciador de filas perdida.

**RC2217**

(2217, X'8A9') Não autorizado para conexão.

**RC2202**

(2202, X'89A') Quiesce de conexão.

**RC2203**

(2203, X'89B') Conexão sendo encerrada.

**RC2207**

(2207, X'89F') Erro de identificador de correlação.

**RC2010**

(2010, X'7DA') Parâmetro de comprimento de dados inválido.

**RC2016**

(2016, X'7E0 ') É inibida para a fila.

**RC2351**

(2351, X'92F') Unidades de trabalho globais conflitam.

**RC2186**

(2186, X'88A') Estrutura de opções de obtenção de mensagem inválida.

**RC2353**

(2353, X'931') Manipulação em uso para unidade de trabalho global.

**RC2018**

(2018, X'7E2') Manipulação de conexões não válida.

**RC2019**

(2019, X'7E3') Manipulação de objetos não válida.

**RC2259**

(2259, X'8D3') Especificação de pesquisa inconsistente.

**RC2245**

(2245, X'8C5') Especificação de unidade de trabalho inconsistente.

**RC2246**

(2246, X'8C6 ') Mensagem sob o cursor inválida para recuperação.

**RC2352**

(2352, X'930') Unidade de trabalho global conflita com unidade de trabalho local.

**RC2247**

(2247, X'8C7') Opções correspondentes inválidas.

- RC2485**  
(2485, X'9B4') Campo *MaxMsgLength* incorreto.
- RC2026**  
(2026, X'7EA') Descritor de mensagem inválido.
- RC2497**  
(2497, X'9C1') O ponto de entrada da função especificada não pôde ser localizado no módulo.
- RC2496**  
(2496, X'9C0') Módulo localizado, no entanto, é do tipo errado; não 32 bits, 64 bits ou uma biblioteca de link dinâmico válida.
- RC2495**  
(2495, X'9BF') Módulo não localizado no caminho da procura ou não autorizado a carregar.
- RC2250**  
(2250, X'8CA') Número de sequência da mensagem inválido.
- RC2331**  
(2331, X'91B') Uso de token da mensagem inválido.
- RC2033**  
(2033, X'7F1') Nenhuma mensagem disponível.
- RC2034**  
(2034, X'7F2') Cursor de navegação não posicionado na mensagem.
- RC2036**  
(2036, X'7F4') Fila não aberta para navegação.
- RC2037**  
(2037, X'7F5') Fila não aberta para entrada.
- RC2041**  
(2041, X'7F9') Definição de objeto alterada desde a abertura.
- RC2101**  
(2101, X'835') Objeto danificado.
- RC2206**  
(2206, X'89E') Código de operação incorreto na chamada API.
- RC2046**  
(2046, X'7FE') Opções não válidas ou não consistentes.
- RC2193**  
(2193, X'891') Erro ao acessar o conjunto de dados do conjunto de páginas.
- RC2052**  
(2052, X'804') A fila foi excluída.
- RC2394**  
(2394, X'95A') Fila possui tipo de índice errado.
- RC2058**  
(2058, X'80A') Nome do gerenciador de filas não válido ou não conhecido.
- RC2059**  
(2059, X'80B') Gerenciador de filas não disponível para conexão.
- RC2161**  
(2161, X'871') Quiesce do gerenciador de filas.
- RC2162**  
(2162, X'872') Gerenciador de filas sendo encerrado.
- RC2102**  
(2102, X'836') Recursos insuficientes do sistema disponíveis.
- RC2069**  
(2069, X'815') Sinais pendentes para essa manipulação.

**RC2071**

(2071, X'817') Armazenamento insuficiente disponível.

**RC2109**

(2109, X'83D') Chamada suprimida pelo programa de saída.

**RC2024**

(2024, X'7E8') Nenhuma mensagem adicional pode ser manipulada dentro da unidade de trabalho atual.

**RC2072**

(2072, X'818 ') Suporte de ponto de sincronização não disponível.

**RC2195**

(2195, X'893') Ocorreu um erro inesperado.

**RC2354**

(2354, X'932') Cadastramento na unidade de trabalho global falhou.

**RC2355**

(2355, X'933') Combinação de chamadas de unidade de trabalho não suportada.

**RC2255**

(2255, X'8CF') Unidade de trabalho não disponível para o gerenciador de filas a usar.

**RC2090**

(2090, X'82A') Intervalo de Espera em MQGMO inválido.

**RC2256**

(2256, X'8D0') Versão errada do MQGMO fornecido.

**RC2257**

(2257, X'8D1') Versão errada do MQMD fornecido.

**RC2298**

(2298, X'8FA') A função solicitada não está disponível no ambiente atual.

**Declaração de RPG**

```

C*..1.....2.....3.....4.....5.....6.....7..
C          CALLP      MQCB(HCONN : OPERATN : CBDSC :
                   HOBJ : MSGDSC : GMO :
                   DATLEN : CMPCOD : REASON)

```

A definição de protótipo para a chamada é:

```

DMQCB          PR          EXTPROC('MQCB')
D* Connection handle
D HCONN          10I 0 VALUE
D* Operation
D OPERATN        10I 0 VALUE
D* Callback descriptor
D CBDSC          180A
D* Object handle
D HOBJ           10I 0 VALUE
D* Message Descriptor
D MSGDSC         364A
D* Get options
D GMO            112A
D* Completion code
D CMPCOD         10I 0
* Reason code qualifying CompCode
D REASON        10I 0

```

**IBM i MQCLOSE (Fechar objeto) no IBM i**

A chamada MQCLOSE renuncia ao acesso a um objeto e é o inverso da chamada MQOPEN

- [“Sintaxe” na página 1314](#)
- [“Observações de Uso” na página 1314](#)
- [“Parâmetros” na página 1315](#)
- [“Declaração de RPG” na página 1320](#)

## Sintaxe

MQCLOSE (*HCONN, HOBJ, OPTS, CMPCOD, REASON*)

## Observações de Uso

1. Quando um aplicativo emite a chamada MQDISC, ou termina normalmente ou de forma anormal, quaisquer objetos que foram abertos pelo aplicativo e ainda estão abertos são fechados automaticamente com a opção CONONE
2. Os pontos a seguir se aplicarão se o objeto que estiver sendo fechado for uma *fila*:
  - Se as operações na fila forem executadas como parte de uma unidade de trabalho, a fila poderá ser fechada antes ou depois que o ponto de sincronização ocorrer sem afetar o resultado do ponto de sincronização.
  - Se a fila foi aberta com a opção OOBW, o cursor de navegação será destruído.. Se a fila for reaberta posteriormente com a opção OBRW, um novo cursor de navegação será criado (consulte a opção OOBW descrita em MQOPEN).
  - Se uma mensagem estiver atualmente bloqueada para esse identificador no momento da chamada MQCLOSE, o bloqueio será liberado (consulte a opção GMLK descrita em [“MQGMO \(opções Get-message\) em IBM i” na página 1113](#) .
3. Os seguintes pontos se aplicam se o objeto que está sendo fechado for uma *fila dinâmica* (permanente ou temporária):
  - Para uma fila dinâmica, as opções CODEL ou COPURG podem ser especificadas, independentemente das opções especificadas na chamada MQOPEN correspondente
  - Quando uma fila dinâmica é excluída, todas as chamadas MQGET com a opção GMWT que estão pendentes na fila são canceladas e o código de razão RC2052 é retornado. Consulte a opção GMWT descrita em [“MQGMO \(opções Get-message\) em IBM i” na página 1113..](#)

Após uma fila dinâmica ter sido excluída, qualquer chamada (diferente de MQCLOSE) que tente fazer referência à fila usando um identificador *HOBJ* adquirido anteriormente falhará com o código de razão RC2052.

Esteja ciente de que, embora uma fila excluída não possa ser acessada por aplicativos, a fila não é removida do sistema e os recursos associados não são liberados até que todos os identificadores que fazem referência à fila tenham sido fechados e todas as unidades de trabalho que afetam a fila tenham sido confirmadas ou restauradas.

  - Quando uma fila dinâmica permanente é excluída, se o identificador de *HOBJ* especificado na chamada MQCLOSE não for aquele que foi retornado pela chamada MQOPEN que criou a fila, será feita uma verificação de que o identificador de usuário que foi usado para validar a chamada MQOPEN está autorizado a excluir a fila.. Se a opção OOALTU foi especificada na chamada MQOPEN, o identificador de usuário marcado é o *ODAU*..

Essa verificação não será executada se:

  - O identificador especificado é aquele retornado pela chamada MQOPEN que criou a fila..
  - A fila sendo excluída é uma fila dinâmica temporária.
  - Quando uma fila dinâmica temporária é fechada, se o identificador *HOBJ* especificado na chamada MQCLOSE for aquele que foi retornado pela chamada MQOPEN que criou a fila, a fila será excluída. Isso ocorre independentemente das opções de fechamento especificadas na chamada MQCLOSE.. Se houver mensagens na fila, elas serão descartadas; nenhuma mensagem de relatório será gerada

Se houver unidades de trabalho não confirmadas que afetam a fila, a fila e suas mensagens ainda serão excluídas, mas isso não causará falha das unidades de trabalho. No entanto, conforme descrito anteriormente, os recursos associados às unidades de trabalho não são liberados até que cada uma delas tenha sido confirmada ou restaurada.

4. Os seguintes pontos se aplicam se o objeto que está sendo fechado for uma *lista de distribuição*:

- A única opção de fechamento válida para uma lista de distribuição é CONONE; a chamada falhará com o código de razão RC2046 ou RC2045 se quaisquer outras opções forem especificadas
- Quando uma lista de distribuição é fechada, os códigos de conclusão individuais e os códigos de razão não são retornados para as filas na lista-somente os parâmetros **CMPCOD** e **REASON** da chamada estão disponíveis para propósitos de diagnóstico

Se ocorrer uma falha ao fechar uma das filas, o gerenciador de filas continua o processamento e tenta fechar as filas restantes na lista de distribuição. Os parâmetros **CMPCOD** e **REASON** da chamada são então configurados para retornar informações que descrevem a falha. Assim, é possível que o código de conclusão seja CCFAIL, mesmo que a maioria das filas tenha sido fechada com sucesso. A fila que encontrou o erro não é identificada.

Se houver uma falha em mais de uma fila, não será definido qual falha será relatada nos parâmetros **CMPCOD** e **REASON**.

## Parâmetros

A chamada MQCLOSE possui os parâmetros a seguir:

### HCONN (número inteiro assinado de 10 dígitos)-entrada

Manipulação de conexões.

Esta manipulação representa a conexão ao gerenciador de filas. O valor de *HCONN* foi retornado por uma chamada anterior MQCONN ou MQCONNX.

### HOBJ (número inteiro assinado de 10 dígitos)-entrada/saída

Manipulação de objetos.

Esse identificador representa o objeto que está sendo fechado. O objeto pode ser de qualquer tipo.. O valor de *HOBJ* foi retornado por uma chamada MQOPEN anterior.

Na conclusão bem-sucedida da chamada, o gerenciador de filas configura esse parâmetro para um valor que não é um identificador válido do ambiente. Esse valor é:.

### HOUNUH

Identificador de objeto não utilizável

### OPTS (número inteiro assinado de 10 dígitos)-entrada

As opções que controlam a ação de MQCLOSE.

O parâmetro **OPTS** controla como o objeto é fechado. Apenas filas e assinaturas dinâmicas permanentes podem ser encerradas de mais de uma maneira. Filas dinâmicas permanentes podem ser retidas ou excluídas; essas são filas com um atributo **DefinitionType** que possui o valor QDPERM (consulte o atributo **DefinitionType** descrito em “Atributos para filas” na página 1416 ). As opções de fechamento são resumidas em uma tabela posteriormente neste tópico.

Assinaturas duráveis podem ser mantidas ou removidas; elas são criadas usando a chamada MQSUB com a opção SODUR.

Ao fechar o identificador para um destino gerenciado (ou seja, o parâmetro **Hobj** retornado em uma chamada MQSUB que usou a opção SOMAN), o gerenciador de filas limpará quaisquer publicações não recuperadas quando a assinatura associada também tiver sido removida. Isso é feito usando a opção CORMSB no parâmetro **Hsub** retornado em uma chamada MQSUB. Observe que CORMSB é o comportamento padrão em MQCLOSE para uma assinatura não durável.

Ao fechar um identificador para um destino não gerenciado, você é responsável por limpar a fila para a qual as publicações são enviadas. Recomenda-se que você feche a assinatura usando CORMSB primeiro e, em seguida, processe as mensagens fora da fila até que não haja nenhum restante..

Um (e apenas um) dos seguintes deve ser especificado:

### Opções de encerramento de fila dinâmica

Estas opções controlam como as filas dinâmicas permanentes são fechadas:

#### CODEL

Exclua a fila.

A fila será excluída se uma das seguintes situações for verdadeira:

- É uma fila dinâmica permanente, criada por uma chamada MQOPEN anterior e não há mensagens na fila e nenhuma solicitação get ou put não confirmada pendente para a fila (para a tarefa atual ou qualquer outra tarefa).
- É a fila dinâmica temporária que foi criada pela chamada MQOPEN que retornou *HOBj*.. Nesse caso, todas as mensagens na fila são limpas

Em todos os outros casos, incluindo o caso em que o *Hobj* foi retornado em uma chamada do MQSUB, a chamada falha com o código de razão RC2045e o objeto não é excluído

#### COPURG

Excluir a fila, limpando qualquer mensagem nela.

A fila será excluída se uma das seguintes situações for verdadeira:

- É uma fila dinâmica permanente, criada por uma chamada MQOPEN anterior e não há solicitações get ou put não confirmadas pendentes para a fila (para a tarefa atual ou qualquer outra tarefa)
- É a fila dinâmica temporária que foi criada pela chamada MQOPEN que retornou *HOBj*..

Em todos os outros casos, incluindo o caso em que o *Hobj* foi retornado em uma chamada do MQSUB, a chamada falha com o código de razão RC2045e o objeto não é excluído

A tabela a seguir mostra quais opções de fechamento são válidas e se o objeto é retido ou excluído

<i>Tabela 743. Opções de fechamento válidas para uso com objetos retidos ou excluídos..</i>			
<b>Tipo de objeto ou fila</b>	<b>CONONE</b>	<b>CODEL</b>	<b>COPURG</b>
Objeto diferente de uma fila	Retida	Inválido	Inválido
Fila predefinida	Retida	Inválido	Inválido
Fila dinâmica permanente	Retida	Excluído se vazio e nenhuma atualização pendente	Mensagens excluídas; fila excluída se nenhuma atualização pendente
Fila dinâmica temporária (chamada emitida pelo criador da fila)	Excluído	Excluído	Excluído
Fila dinâmica temporária (chamada não emitida pelo criador da fila)	Retida	Inválido	Inválido
Lista de distribuição	Retida	Inválido	Inválido
Destino de assinatura gerenciado	Retida	Inválido	Inválido



<i>Tabela 743. Opções de fechamento válidas para uso com objetos retidos ou excluídos.. (continuação)</i>			
<b>Tipo de objeto ou fila</b>	<b>CONONE</b>	<b>CODEL</b>	<b>COPURG</b>
Lista de distribuição (a assinatura foi removida)	Mensagens excluídas; fila excluída	Inválido	Inválido

### **Opções de encerramento de assinatura**

Essas opções controlam se as assinaturas duráveis serão removidas quando a manipulação for fechada e se as publicações ainda aguardando para serem lidas pelo aplicativo serão limpas. Essas opções são válidas somente para uso com uma manipulação de objetos retornada no parâmetro **HSUB** de uma chamada MQSUB

#### **COKPSB**

O identificador para a assinatura é fechado, mas a assinatura feita é mantida. As publicações continuarão sendo enviadas para o destino especificado na assinatura. Essa opção será válida apenas se a assinatura tiver sido feita com a opção SODUR COKPSB será o padrão se a assinatura for durável

#### **CORMSB**

A assinatura é removida e a manipulação para a assinatura é fechada

O parâmetro **Hobj** da chamada MQSUB não é invalidado pelo encerramento do parâmetro **Hsub** e pode continuar sendo usado para MQGET ou MQCB receber as publicações restantes. Quando o parâmetro **Hobj** da chamada MQSUB também for fechado, se ele for um destino gerenciado, quaisquer publicações não recuperadas serão removidas

CORMSB é o padrão se a subscrição não for durável

Essas opções de encerramento de assinatura são resumidas nas tabelas a seguir:

Para fechar uma manipulação de assinatura durável, mas deixar a assinatura em torno, use as seguintes opções de encerramento de assinatura:

<i>Tabela 744. Opções de tarefa para fechar um identificador de assinatura durável e deixar a assinatura ao redor</i>	
<b>Tarefa</b>	<b>Opção de encerramento de assinatura</b>
Manter publicações em um identificador MQOPENed	COKPSB
Remover publicações em um identificador MQOPENed	Ação não permitida
Mantenha as publicações em um identificador com o SOMAN	COKPSB
Remover publicações em um identificador com SOMAN	Ação não permitida

Para cancelar a assinatura, fechando um identificador de assinatura durável e cancelando a assinatura ou fechando um identificador de assinatura não durável, use as seguintes opções de fechamento de assinatura:

<i>Tabela 745. Opções de tarefa para cancelar assinatura</i>	
<b>Tarefa</b>	<b>Opção de encerramento de assinatura</b>
Manter publicações em um identificador MQOPENed	CORMSB
Remover publicações em um identificador MQOPENed	Ação não permitida
Mantenha as publicações em um identificador com o SOMAN	CORMSB

Tabela 745. Opções de tarefa para cancelar assinatura (continuação)	
Tarefa	Opção de encerramento de assinatura
Remover publicações em um identificador com SOMAN	COPGSB

### Opções de leitura antecipada

As opções a seguir controlam o que acontece com mensagens não persistentes que foram enviadas ao cliente antes que um aplicativo as solicitasse e ainda não foram consumidas pelo aplicativo. Essas mensagens são armazenadas no buffer de leitura antecipada do cliente aguardando para serem solicitadas pelo aplicativo e podem ser descartadas ou consumidas da fila antes que o MQCLOSE seja concluído

#### COIMM

O objeto é fechado imediatamente e quaisquer mensagens que foram enviadas ao cliente antes de um aplicativo as solicitar são descartadas e não estão disponíveis para serem consumidas por nenhum aplicativo. Esse é o valor-padrão.

#### COQSC

Uma solicitação para fechar o objeto é feita, mas se quaisquer mensagens que foram enviadas ao cliente antes que um aplicativo as solicitasse, ainda residirem no buffer de leitura antecipada do cliente, a chamada MQCLOSE retornará com um código de aviso de RC2458e a manipulação de objetos permanecerá válida..

O aplicativo pode, então, continuar a usar o identificador de objetos para recuperar mensagens até que não mais estejam disponíveis e, em seguida, fechar o objeto novamente. Nenhuma outra mensagem será enviada para o cliente antes de um pedido de aplicativo, então, a leitura antecipada está desativada.

Os aplicativos são aconselhados a usar COQSC em vez de tentar atingir um ponto em que não haja mais mensagens no buffer de leitura antecipada do cliente, já que uma mensagem poderia chegar entre a última chamada MQGET e o MQCLOSE a seguir que seria descartado se COIMM fosse usado.

Se um MQCLOSE com COQSC for emitido de dentro de uma função de retorno de chamada assíncrona, o mesmo comportamento de leitura antecipada de mensagens será aplicado. Se o código de aviso RC2458 for retornado, a função de retorno de chamada será chamada pelo menos mais uma vez. Quando a última mensagem restante que foi lida antecipadamente foi transmitida para a função de retorno de chamada, o campo CBCFLG é definido como CBCFBE.

### Opção Padrão

Se você não precisar de nenhuma das opções descritas anteriormente, poderá usar a opção a seguir:

#### CONONE

Nenhum processamento de fechamento opcional necessário.

Isso deve ser especificado para:

- Objetos diferentes de filas
- Filas Predefinidas
- Filas dinâmicas temporárias (mas somente nos casos em que *HOBJ* não é o identificador retornado pela chamada MQOPEN que criou a fila).
- Listas de distribuição

Em todos os casos anteriores, o objeto é mantido e não excluído.

Se esta opção for especificada para uma fila dinâmica temporária:

- A fila será excluída, se tiver sido criada pela chamada MQOPEN que retornou *HOBJ* ; todas as mensagens que estão na fila são limpas
- Em todos os outros casos, a fila (e quaisquer mensagens nela) são retidos..

Se essa opção for especificada para uma fila dinâmica permanente a fila será retida e não excluída.

### **CMPCOD (número inteiro assinado de 10 dígitos)-saída**

Código de conclusão.

É um dos seguintes:

#### **CCOK**

Indica conclusão bem-sucedida.

#### **CCWARN**

Aviso (conclusão parcial).

#### **CCFAIL**

A chamada falhou.

### **REASON (número inteiro assinado de 10 dígitos)-saída**

Código de razão qualificando *CMPCOD*.

Se *CMPCOD* for CCOK:

#### **RCNONE**

(0, X'000') Nenhuma razão a relatar.

Se *CMPCOD* for CCWARN:

#### **RC2241**

(2241, X'8C1') Grupo de mensagens não concluído.

#### **RC2242**

(2242, X'8C2') Mensagem lógica não concluída.

Se *CMPCOD* for CCFAIL:

#### **RC2219**

(2219, X'8AB') chamada MQI reinserida antes da chamada anterior ser concluída.

#### **RC2009**

(2009, X'7D9') Conexão ao gerenciador de filas perdida.

#### **RC2018**

(2018, X'7E2') Manipulação de conexões não válida.

#### **RC2019**

(2019, X'7E3') Manipulação de objetos não válida.

#### **RC2035**

(2035, X'7F3') Não autorizado para acesso.

#### **RC2101**

(2101, X'835') Objeto danificado.

#### **RC2045**

(2045, X'7FD') Opção não válida para tipo de objeto.

#### **RC2046**

(2046, X'7FE') Opções não válidas ou não consistentes.

#### **RC2058**

(2058, X'80A') Nome do gerenciador de filas não válido ou não conhecido.

#### **RC2059**

(2059, X'80B') Gerenciador de filas não disponível para conexão.

#### **RC2162**

(2162, X'872') Gerenciador de filas sendo encerrado.

#### **RC2055**

(2055, X'807 ') A fila contém uma ou mais mensagens ou solicitações put ou get não confirmados.

**RC2102**

(2102, X'836') Recursos insuficientes do sistema disponíveis.

**RC2063**

(2063, X'80F') Ocorreu um erro na segurança.

**RC2071**

(2071, X'817') Armazenamento insuficiente disponível.

**RC2195**

(2195, X'893') Ocorreu um erro inesperado.

**Declaração de RPG**

```

C*.1.....2.....3.....4.....5.....6.....7..
C          CALLP      MQCLOSE(HCONN : HOBJ : OPTS :
C                               CMPCOD : REASON)

```

A definição de protótipo para a chamada é:

```

D*.1.....2.....3.....4.....5.....6.....7..
DMQCLOSE      PR          EXTPROC('MQCLOSE')
D* Connection handle
D HCONN              10I 0 VALUE
D* Object handle
D HOBJ              10I 0
D* Options that control the action of MQCLOSE
D OPTS              10I 0 VALUE
D* Completion code
D CMPCOD            10I 0
D* Reason code qualifying CMPCOD
D REASON            10I 0

```

**IBM i MQCMIT (Confirmar mudanças) no IBM i**

A chamada MQCMIT indica ao gerenciador de filas que o aplicativo atingiu um ponto de sincronização e que todas as obtenções e colocações de mensagens que ocorreram desde o último ponto de sincronização devem se tornar permanentes. Mensagens colocadas como parte de uma unidade de trabalho são disponibilizadas para outros aplicativos; mensagens recuperadas como parte de uma unidade de trabalho são excluídas.

- [“Sintaxe” na página 1320](#)
- [“Observações de Uso” na página 1320](#)
- [“Parâmetros” na página 1321](#)
- [“Declaração de RPG” na página 1322](#)

**Sintaxe**

MQCMIT (HCONN, COMCOD, REASON)

**Observações de Uso**

Considere estas notas de uso ao usar MQCMIT..

1. Essa chamada pode ser usada apenas quando o próprio gerenciador de filas coordena a unidade de trabalho. Esta é uma unidade de trabalho local, em que as alterações afetam apenas recursos IBM MQ.
2. Em ambientes em que o gerenciador de filas não coordena a unidade de trabalho, a chamada de confirmação apropriada deve ser usada em vez de MQCMIT. O ambiente também pode suportar uma confirmação implícita causada pela finalização do aplicativo normalmente.

- No IBM i, essa chamada pode ser usada para unidades locais de trabalho coordenadas pelo gerenciador de fila.. Isso significa que uma definição de confirmação não deve existir no nível da tarefa, ou seja, o comando STRCMTCTL com o parâmetro **CMTSCOPE (\*JOB)** não deve ter sido emitido para a tarefa
3. Se um aplicativo terminar com mudanças não confirmados em uma unidade de trabalho, a disposição dessas mudanças dependerá se o aplicativo será encerrado normalmente ou de forma anormal. Consulte as notas de uso em [“MQDISC \(Desconectar o gerenciador de filas\) no IBM i”](#) na página 1336 para obter detalhes adicionais
  4. Quando um aplicativo coloca ou obtém mensagens em grupos ou segmentos de mensagens lógicas, o gerenciador de filas retém informações relacionadas ao grupo de mensagens e mensagens lógicas para as últimas chamadas MQPUT e MQGET bem-sucedidas. Essas informações estão associadas ao identificador de filas e incluem itens como:
    - Os valores dos campos *MDGID*, *MDSEQ*, *MDOFF* e *MDMFL* em MQMD..
    - Indica se a mensagem faz parte de uma unidade de trabalho
    - Para a chamada MQPUT: se a mensagem é persistente ou não persistente.

Quando uma unidade de trabalho é confirmada, o gerenciador de filas retém as informações do grupo e do segmento e o aplicativo pode continuar colocando ou obtendo mensagens no grupo de mensagens ou na mensagem lógica atual

Retar as informações do grupo e do segmento quando uma unidade de trabalho é confirmada permite que o aplicativo difunda um grupo de mensagens grandes ou uma mensagem lógica grande que consiste em muitos segmentos em várias unidades de trabalho... Usar várias unidades de trabalho pode ser vantajoso se o gerenciador de fila local tiver apenas armazenamento de fila limitado. No entanto, o aplicativo deve manter informações suficientes para poder reiniciar a colocação ou obtenção de mensagens no ponto correto se ocorrer uma falha do sistema. Para obter detalhes sobre como reiniciar no ponto correto depois de uma falha do sistema, consulte a opção PMLOGO descrita em [“MQPMO \(opções Put-message\) em IBM i”](#) na página 1215e a opção GMLOGO descrita em [“MQGMO \(opções Get-message\) em IBM i”](#) na página 1113

As notas de uso restantes se aplicam apenas quando o gerenciador de filas coordena as unidades de trabalho:

1. Uma unidade de trabalho tem o mesmo escopo que uma manipulação de conexões.. Isso significa que todas as chamadas do IBM MQ que afetam uma unidade de trabalho específica devem ser executadas usando o mesmo identificador de conexões. Chamadas emitidas usando um identificador de conexão diferente (por exemplo, chamadas emitidas por outro aplicativo) afetam uma unidade de trabalho diferente. Consulte o parâmetro **HCONN** descrito em MQCONN para obter informações sobre o escopo de identificadores de conexão
2. Apenas as mensagens que foram colocadas ou recuperadas como parte da unidade de trabalho atual são afetada por esta chamada
3. Um aplicativo de longa execução que emite chamadas MQGET, MQPUT ou MQPUT1 dentro de uma unidade de trabalho, mas que nunca emite uma chamada de confirmação ou de retorno, pode fazer com que as filas sejam preenchidas com mensagens que não estão disponíveis para outros aplicativos. Para proteger contra essa possibilidade, o administrador deve configurar o atributo do gerenciador de filas **MaxUncommittedMsgs** para um valor que seja baixo o suficiente para evitar que aplicativos runaway preencham as filas, mas alto o suficiente para permitir que os aplicativos do sistema de mensagens esperados funcionem corretamente.

## Parâmetros

A chamada MQCMIT possui os parâmetros a seguir:

### **HCONN (número inteiro assinado de 10 dígitos)-entrada**

Manipulação de conexões.

Esta manipulação representa a conexão ao gerenciador de filas. O valor de *HCONN* foi retornado por uma chamada anterior MQCONN ou MQCONNX.

## COMCOD (número inteiro assinado de 10 dígitos)-saída

Código de conclusão.

É um dos seguintes:

### CCOK

Indica conclusão bem-sucedida.

### CCWARN

Aviso (conclusão parcial).

### CCFAIL

A chamada falhou.

## REASON (número inteiro assinado de 10 dígitos)-saída

Código de razão qualificando *COMCOD*.

Se *COMCOD* for CCOK:

### RCNONE

(0, X'000') Nenhuma razão a relatar.

Se *COMCOD* for CCWARN:

### RC2003

(2003, X'7D3') Unidade de trabalho retornada.

### RC2124

(2124, X'84C') Resultado da operação de confirmação pendente.

Se *COMCOD* for CCFAIL:

### RC2219

(2219, X'8AB') chamada MQI reinserida antes da chamada anterior ser concluída.

### RC2009

(2009, X'7D9') Conexão ao gerenciador de filas perdida.

### RC2018

(2018, X'7E2') Manipulação de conexões não válida.

### RC2101

(2101, X'835') Objeto danificado.

### RC2123

(2123, X'84B') O resultado da operação de confirmação ou de retorno é combinado.

### RC2162

(2162, X'872') Gerenciador de filas sendo encerrado.

### RC2102

(2102, X'836') Recursos insuficientes do sistema disponíveis.

### RC2071

(2071, X'817') Armazenamento insuficiente disponível.

### RC2195

(2195, X'893') Ocorreu um erro inesperado.

## Declaração de RPG

```
C*..1.....2.....3.....4.....5.....6.....7..  
C          CALLP      MQCMIT(HCONN : COMCOD : REASON)
```

A definição de protótipo para a chamada é:

```
D*..1.....2.....3.....4.....5.....6.....7..  
DMQCMIT      PR          EXTPROC('MQCMIT')
```

```
D* Connection handle
D HCONN                10I 0 VALUE
D* Completion code
D COMCOD              10I 0
D* Reason code qualifying COMCOD
D REASON              10I 0
```

## IBM i MQCONN (Conectar gerenciador de filas) no IBM i

A chamada MQCONN conecta um programa de aplicativo a um gerenciador de filas. Ele fornece um identificador de conexão do gerenciador de fila, que é usado pelo aplicativo em chamadas de enfileiramento de mensagem subsequentes

- Os aplicativos devem usar a chamada MQCONN ou MQCONNX para se conectar ao gerenciador de fila e a chamada MQDISC para se desconectar do gerenciador de filas.

No IBM MQ for Multiplatforms, cada encadeamento em um aplicativo pode se conectar a gerenciadores de filas diferentes. Em outros sistemas, todas as conexões simultâneas em um processo devem estar no mesmo gerenciador de filas.

- [“Sintaxe” na página 1323](#)
- [“Observações de Uso” na página 1323](#)
- [“Parâmetros” na página 1324](#)
- [“Declaração de RPG” na página 1326](#)

### Sintaxe

MQCONN (QMNAME, HCONN, CMPCOD, REASON)

### Observações de Uso

1. O gerenciador de filas ao qual a conexão é feita usando a chamada MQCONN é chamado de *gerenciador de filas locais*.
2. As filas pertencentes ao gerenciador de filas locais aparecem para o aplicativo como filas locais. É possível colocar mensagens e obter mensagens dessas filas.  
  
Filas compartilhadas que pertencem ao grupo de filas compartilhadas ao qual o gerenciador de filas locais pertence aparecem para o aplicativo como filas locais. É possível colocar mensagens e obter mensagens dessas filas.  
  
As filas pertencentes a gerenciadores de filas remotas aparecem como filas remotas. É possível colocar mensagens nessas filas, mas não é possível obter mensagens dessas filas.
3. Se o gerenciador de filas falhar enquanto um aplicativo estiver em execução, o aplicativo deverá emitir a chamada MQCONN novamente para obter uma nova manipulação de conexões para usar em chamadas IBM MQ subsequentes. O aplicativo pode emitir a chamada MQCONN periodicamente, até que ela seja bem-sucedida.  
  
Se um aplicativo não tiver certeza se está conectado ao gerenciador de filas, o aplicativo poderá emitir com segurança uma chamada MQCONN para obter uma manipulação de conexões. Se o aplicativo já estiver conectado, o identificador retornado será o mesmo retornado pela chamada MQCONN anterior, mas com o código de conclusão CCWARN e o código de razão RC2002.
4. Quando o aplicativo tiver concluído o uso de chamadas IBM MQ, o aplicativo deverá usar a chamada MQDISC para desconectar do gerenciador de filas.
5. No IBM i, os programas que terminam de forma anormal não são desconectados automaticamente do gerenciador de filas. Portanto, os aplicativos devem ser gravados para permitir a possibilidade de a chamada MQCONN ou MQCONNX retornar o código de conclusão CCWARN e o código de razão RC2002. A manipulação de conexões retornada nesta situação pode ser usada normalmente

## Parâmetros

A chamada MQCONN possui os seguintes parâmetros:

### QMNAME (sequência de caracteres de 48 bytes)-entrada de

Nome do gerenciador de filas.

Este é o nome do gerenciador de filas ao qual o aplicativo deseja se conectar. O nome pode conter os seguintes caracteres:

- Caracteres alfabéticos maiúsculos (A a Z)
- Caracteres alfabéticos minúsculos (a a z)
- Dígitos numéricos (0 a 9)
- Ponto (.), barra (/), sublinhado (\_), porcentagem (%)

O nome não deve conter espaços em branco iniciais ou integrados, mas pode conter espaços em branco finais. Um caractere nulo pode ser usado para indicar o término de dados significativos no nome; o nulo e quaisquer caracteres subsequentes são tratados como espaços em branco. As restrições a seguir aplicam-se aos ambientes indicados:

- No IBM i, os nomes contendo caracteres minúsculos, barra ou porcentagem devem ser colocados entre aspas quando especificados em comandos. Essas aspas não devem ser especificadas no parâmetro **QMNAME**

Se o nome consistir inteiramente de espaços em branco, o nome do gerenciador de filas *padrão* será usado.

O nome especificado para *QMNAME* deve ser o nome de um gerenciador de filas *conectável* ..

**Grupos de filas compartilhadas:** Em sistemas em que existem vários gerenciadores de filas e são configurados para formar um grupo de filas compartilhadas, o nome do grupo de filas compartilhadas pode ser especificado para *QMNAME* no lugar do nome de um gerenciador de filas. Isso permite que o aplicativo se conecte a *qualquer* gerenciador de fila disponível no grupo de filas compartilhadas. O sistema também pode ser configurado para que um *QMNAME* em branco cause conexão com o grupo de filas compartilhadas em vez do gerenciador de filas padrão.

Se *QMNAME* especificar o nome do grupo de filas compartilhadas, mas também houver um gerenciador de fila com esse nome no sistema, a conexão será feita com o último em preferência ao primeiro... Somente se essa conexão falhar, será feita uma tentativa de conexão com um dos gerenciadores de filas no grupo de filas compartilhadas.

Se a conexão for bem-sucedida, a manipulação retornada pela chamada MQCONN ou MQCONNX poderá ser usada para acessar *todos* os recursos (compartilhados e não compartilhados) que pertencem ao gerenciador de filas específico para o qual a conexão foi feita. O acesso a esses recursos está sujeito aos controles típicos de autorização.

Se o aplicativo emitir duas chamadas MQCONN ou MQCONNX para estabelecer conexões simultâneas e uma ou ambas especificar o nome do grupo de filas compartilhadas, a segunda chamada poderá retornar o código de conclusão CCWARN e o código de razão RC2002. Isso ocorre quando a segunda chamada se conecta ao mesmo gerenciador de filas da primeira chamada.

Grupos de filas compartilhadas são suportados somente no z/OS. A conexão com um grupo de filas compartilhadas é suportada apenas nos ambientes em lote, RRS e TSO.

**IBM MQ aplicativos clientes:** Para aplicativos IBM MQ MQI client, uma conexão é tentada para cada definição de canal de conexão do cliente com o nome do gerenciador de filas especificado, até que uma seja bem-sucedida. O gerenciador de filas, no entanto, deve ter o mesmo nome que o nome especificado. Se um nome em branco for especificado, cada canal de conexão do cliente com um nome do gerenciador de filas em branco será tentado até que um seja bem-sucedido; neste caso, não há verificação com relação ao nome real do gerenciador de filas.

**IBM MQ grupos de gerenciadores de filas do cliente:** se o nome especificado iniciar com um asterisco (\*), o gerenciador de fila real com o qual a conexão é feita poderá ter um nome diferente



daquele especificado pelo aplicativo. O nome especificado (sem o asterisco) define um *grupo* de gerenciadores de filas que são elegíveis para conexão. A implementação seleciona um do grupo tentando cada um por vez, em ordem alfabética, até que um seja localizado para o qual uma conexão pode ser feita. Se nenhum dos gerenciadores de filas no grupo estiver disponível para conexão, a chamada falhará. Cada gerenciador de filas é tentando somente uma vez. Se um asterisco sozinho for especificado para o nome, um grupo de gerenciadores de filas padrão definido pela implementação será usado

Os grupos de gerenciadores de filas são suportados apenas para aplicativos em execução em um ambiente do cliente MQ; a chamada falhará se um aplicativo não cliente especificar um nome do gerenciador de filas que comece com um asterisco. Um grupo é definido fornecendo várias definições de canal de conexão do cliente com o mesmo nome do gerenciador de filas (o nome especificado sem o asterisco) para se comunicar com cada um dos gerenciadores de filas no grupo. O grupo padrão é definido fornecendo uma ou mais definições de canal de conexão do cliente, cada uma com um nome de gerenciador de filas em branco (especificar um nome todo em branco, portanto, tem o mesmo efeito que especificar um único asterisco para o nome de um aplicativo cliente)

Depois de se conectar a um gerenciador de fila de um grupo, um aplicativo pode especificar espaços em branco da maneira típica nos campos de nome do gerenciador de filas nos descritores de mensagem e objeto para significar o nome do gerenciador de filas ao qual o aplicativo foi realmente conectado (o *gerenciador de fila local*). Se o aplicativo precisar saber esse nome, a chamada MQINQ poderá ser emitida para consultar o atributo do gerenciador de filas do **QMgrName**

Prefixar um asterisco para o nome da conexão implica que o aplicativo não é dependente da conexão com um gerenciador de filas específico no grupo. As aplicações adequadas seriam:

- Aplicativos que colocam mensagens, mas não obtêm mensagens.
- Aplicativos que colocam mensagens de solicitação e, em seguida, obtêm as mensagens de resposta de uma fila *dinâmica temporária*.

Os aplicativos inadequados seriam aqueles que precisam obter mensagens de uma fila específica em um gerenciador de filas específico; esses aplicativos não devem prefixar o nome com um asterisco

Observe que, se um asterisco for especificado, o comprimento máximo do restante do nome será de 47 caracteres.

O comprimento deste parâmetro é fornecido por LNQMNM

### **HCONN (número inteiro assinado de 10 dígitos)-saída**

Manipulação de conexões.

Esta manipulação representa a conexão ao gerenciador de filas. Ele deve ser especificado em todas as chamadas de fila de mensagens subsequentes emitidas pelo aplicativo. Ela deixa de ser válida quando a chamada MQDISC é emitida ou quando a unidade de processamento que define o escopo da manipulação é finalizada.

O escopo da alça é restrito à menor unidade de processamento paralelo suportado pela plataforma na qual o aplicativo está em execução; o identificador não é válido fora da unidade de processamento paralelo da qual a chamada MQCONN foi emitida.

- No IBM i, o escopo do identificador é a tarefa que emite a chamada.

### **CMPCOD (número inteiro assinado de 10 dígitos)-saída**

Código de conclusão.

É um dos seguintes:

#### **CCOK**

Indica conclusão bem-sucedida.

#### **CCWARN**

Aviso (conclusão parcial).

#### **CCFAIL**

A chamada falhou.

## REASON (número inteiro assinado de 10 dígitos)-saída

Código de razão qualificando *CMPCOD*.

Se *CMPCOD* for CCOK:

### RCNONE

(0, X'000') Nenhuma razão a relatar.

Se *CMPCOD* for CCWARN:

### RC2002

(2002, X'7D2') Aplicativo já conectado.

Se *CMPCOD* for CCFAIL:

### RC2219

(2219, X'8AB') chamada MQI reinserida antes da chamada anterior ser concluída.

### RC2267

(2267, X'8DB') Não é possível carregar a saída de carga de trabalho do cluster.

### RC2009

(2009, X'7D9') Conexão ao gerenciador de filas perdida.

### RC2018

(2018, X'7E2') Manipulação de conexões não válida.

### RC2035

(2035, X'7F3') Não autorizado para acesso.

### RC2137

(2137, X'859') Objeto não aberto com sucesso.

### RC2058

(2058, X'80A') Nome do gerenciador de filas não válido ou não conhecido.

### RC2059

(2059, X'80B') Gerenciador de filas não disponível para conexão.

### RC2161

(2161, X'871') Quiesce do gerenciador de filas.

### RC2162

(2162, X'872') Gerenciador de filas sendo encerrado.

### RC2102

(2102, X'836') Recursos insuficientes do sistema disponíveis.

### RC2063

(2063, X'80F') Ocorreu um erro na segurança.

### RC2071

(2071, X'817') Armazenamento insuficiente disponível.

### RC2195

(2195, X'893') Ocorreu um erro inesperado.

## Declaração de RPG

```
C*..1.....2.....3.....4.....5.....6.....7..
C          CALLP      MQCONN(QMNAME : HCONN : CMPCOD :
C                                REASON)
```

A definição de protótipo para a chamada é:

```
D*..1.....2.....3.....4.....5.....6.....7..
DMQCONN          PR          EXTPROC('MQCONN')
D* Name of queue manager
D QMNAME          48A
D* Connection handle
```

D HCONN	10I 0
D* Completion code	
D CMPCOD	10I 0
D* Reason code qualifying CMPCOD	
D REASON	10I 0

## IBM i MQCONNX (Conectar gerenciador de filas (estendido)) no IBM i

A chamada MQCONNX conecta um programa aplicativo a um gerenciador de filas. Ele fornece uma manipulação de conexões do gerenciador de filas, que é usada pelo aplicativo em chamadas subsequentes do IBM MQ.

A chamada MQCONNX é como a chamada MQCONN, exceto que MQCONNX permite que opções sejam especificadas para controlar a maneira como a chamada funciona.

No IBM MQ for Multiplatforms, cada encadeamento em um aplicativo pode se conectar a gerenciadores de filas diferentes. Em outros sistemas, todas as conexões simultâneas em um processo devem estar no mesmo gerenciador de filas.

- [“Sintaxe” na página 1327](#)
- [“Parâmetros” na página 1327](#)
- [“Declaração de RPG” na página 1328](#)

### Sintaxe

MQCONNX (*QMNAME*, *CNOPT*, *HCONN*, *CMPCOD*, *REASON*)

### Parâmetros

A chamada MQCONNX possui os parâmetros a seguir:

#### QMNAME (sequência de caracteres de 48 bytes)-entrada de

Nome do gerenciador de filas.

Consulte o parâmetro **QMNAME** descrito em [“MQCONN \(Conectar gerenciador de filas\) no IBM i” na página 1323](#) para obter detalhes..

#### CNOPT (MQCNO)-entrada/saída

Opções que controlam a ação de MQCONNX.

Consulte [“MQCNO \(Opções de Conexão\) no IBM i” na página 1083](#) para obter detalhes.

#### HCONN (número inteiro assinado de 10 dígitos)-saída

Manipulação de conexões.

Consulte o parâmetro **HCONN** descrito em [“MQCONN \(Conectar gerenciador de filas\) no IBM i” na página 1323](#) para obter detalhes..

#### CMPCOD (número inteiro assinado de 10 dígitos)-saída

Código de conclusão.

Consulte o parâmetro **CMPCOD** descrito em [“MQCONN \(Conectar gerenciador de filas\) no IBM i” na página 1323](#) para obter detalhes..

#### REASON (número inteiro assinado de 10 dígitos)-saída

Código de razão qualificando *CMPCOD*.

Consulte o parâmetro **REASON** descrito em [“MQCONN \(Conectar gerenciador de filas\) no IBM i” na página 1323](#) para obter detalhes de possíveis códigos de razão.

Os seguintes códigos de razão adicionais podem ser retornados pela chamada MQCONNX:

Se *CMPCOD* for CCFAIL:

#### **RC2278**

(2278, X'8E6') Campos de conexão do cliente não válidos.

#### **RC2139**

(2139, X'85B') Estrutura de opções de conexão não válida.

#### **RC2046**

(2046, X'7FE') Opções não válidas ou não consistentes.

## Declaração de RPG

```
C*..1.....2.....3.....4.....5.....6.....7..
C                               CALLP      MQCONN(QMNAME : HCONN : CMPCOD :
C                               REASON)
```

A definição de protótipo para a chamada é:

```
D*..1.....2.....3.....4.....5.....6.....7..
DMQCONN      PR      EXTPROC('MQCONN')
D* Name of queue manager
D QMNAME      48A
D* Options that control the action of MQCONN
D HCONN      224A
D* Connection handle
D HCONN      10I 0
D* Completion code
D CMPCOD      10I 0
D* Reason code qualifying CMPCOD
D REASON      10I 0
```

## MQCRTMH (Criar identificador de mensagem) no IBM i

A chamada de MQCRTMH retorna uma manipulação de mensagens

Um aplicativo pode usá-lo em chamadas subsequentes de enfileiramento de mensagens:

- Use a chamada [MQSETMP](#) para configurar uma propriedade do identificador de mensagem
- Use a chamada [MQINQMP](#) para consultar o valor de uma propriedade do identificador de mensagem
- Use a chamada [MQDLTMP](#) para excluir uma propriedade da manipulação de mensagens

A manipulação de mensagem pode ser usada nas chamadas MQPUT e MQPUT1 para associar as propriedades da manipulação de mensagem às propriedades da mensagem que está sendo colocada. Da mesma forma, ao especificar um identificador de mensagens na chamada MQGET, as propriedades da mensagem que está sendo recuperada podem ser acessadas usando o identificador de mensagens quando a chamada MQGET for concluída

Use [MQDLTMH](#) para excluir o identificador da mensagem

- [“Sintaxe” na página 1328](#)
- [“Parâmetros” na página 1328](#)
- [“Declaração de RPG” na página 1330](#)

### Sintaxe

MQCRTMH (*Hconn*, *CrtMsgHOpts*, *Hmsg*, *CompCode*, *Reason*)

### Parâmetros

A chamada MQCRTMH possui os parâmetros a seguir:

## **HCONN (número inteiro assinado de 10 dígitos)-entrada**

Esta manipulação representa a conexão ao gerenciador de filas. O valor de *HCONN* foi retornado por uma chamada anterior *MQCONN* ou *MQCONNX*. Se a conexão com o gerenciador de filas deixar de ser válida e nenhuma chamada IBM MQ estiver operando na manipulação de mensagens, *MQDLTMH* será chamado implicitamente para excluir a mensagem

Como alternativa, é possível especificar o seguinte valor:

### **HCUNAS**

A manipulação de conexões não representa uma conexão com nenhum gerenciador de fila específico

Quando esse valor for usado, o identificador de mensagens deverá ser excluído com uma chamada explícita para *MQDLTMH* para liberar qualquer armazenamento alocado para ele. IBM MQ nunca exclui implicitamente o identificador de mensagem..

Deve haver pelo menos uma conexão válida para um gerenciador de filas estabelecido no encadeamento que está criando a manipulação de mensagens, caso contrário, a chamada falhará com RC2018

## **CRTOPT (MQCMHO)-entrada**

As opções que controlam a ação de *MQCRTMH* Consulte *MQCMHO* para obter detalhes..

## **HMSG (número inteiro assinado de 20 dígitos)-saída**

Na saída, é retornada uma manipulação de mensagem que pode ser usada para configurar, consultar e excluir propriedades da manipulação de mensagens Inicialmente, a manipulação de mensagem não contém propriedades

Um identificador de mensagens também possui um descritor de mensagens associado Inicialmente, esse descritor de mensagem contém os valores padrão Os valores dos campos do descritor de mensagens associados podem ser configurados e consultados usando as chamadas *MQSETMP* e *MQINQMP* A chamada *MQDLTMP* reconfigura um campo do descritor de mensagens para seu valor padrão.

Se o parâmetro *HCONN* for especificado como o valor *HCUNAS*, o identificador de mensagem retornado poderá ser usado em chamadas *MQGET*, *MQPUT* ou *MQPUT1* com qualquer conexão dentro da unidade de processamento, mas poderá estar em uso por apenas uma chamada IBM MQ por vez. Se o identificador estiver em uso quando uma segunda chamada IBM MQ tentar usar o mesmo identificador de mensagem, a segunda chamada IBM MQ falhará com o código de razão RC2499.

Se o parâmetro *HCONN* não for *HCUNAS*, a manipulação de mensagem retornada poderá ser usada apenas na conexão especificada

O mesmo valor de parâmetro *HCONN* deve ser usado nas chamadas *MQI* subsequentes em que esse identificador de mensagens é usado:

- *MQDLTMH*
- *MQSETMP*
- *MQINQMP*
- *MQDLTMP*
- *MQMHBUF*
- *MQBUFMH*

A manipulação de mensagem retornada deixa de ser válida quando a chamada *MQDLTMH* é emitida para a manipulação de mensagens ou quando a unidade de processamento que define o escopo da manipulação termina. *MQDLTMH* será chamado implicitamente se uma conexão específica for fornecida quando a manipulação de mensagens for criada e a conexão com o gerenciador de fila deixar de ser válida, por exemplo, se o *MQDBC* for chamado

## CMPCOD (número inteiro assinado de 10 dígitos)-saída

O código de conclusão; é um dos seguintes:

### CCOK

Indica conclusão bem-sucedida.

### CCFAIL

A chamada falhou.

## REASON (número inteiro assinado de 10 dígitos)-saída

O código de razão qualificando *CMPCOD*.

Se *CMPCOD* for CCOK:

### RCNONE

(0, X'000') Nenhuma razão a relatar.

Se *CMPCOD* for CCFAIL:

### RC2204

(2204, X'089C') Adaptador não disponível.

### RC2130

(2130, X'852') Não foi possível carregar o módulo de serviço do adaptador.

### RC2157

(2157, X'86D') ASIDs de início e primárias são diferentes.

### RC2219

(2219, X'08AB') chamada MQI inserida antes da chamada anterior ser concluída.

### RC2461

(2461, X'099D') A estrutura de opções de manipulação de mensagens não é válida.

### RC2273

(2273, X'7D9') Conexão com o gerenciador de filas perdida.

### RC2017

(2017, X'07E1') Mais nenhum identificador disponível.

### RC2018

(2018, X'7E2') Manipulação de conexões não válida.

### RC2460

(2460, X'099C') Ponteiro do identificador de mensagens inválido.

### RC2046

(2046, X'07FE') Opções não válidas ou não consistentes.

### RC2071

(2071, X'817') Armazenamento insuficiente disponível.

### RC2195

(2195, X'893') Ocorreu um erro inesperado.

Consulte [“Códigos de retorno para IBM i \(ILE RPG\)”](#) na página 1477 para obter mais detalhes.

## Declaração de RPG

```
C*..1.....2.....3.....4.....5.....6.....7..
C          CALLP      MQCRTMH(HCONN : CRTOPT : HMSG :
                        CMPCOD : REASON)
```

A definição de protótipo para a chamada é:

```
DMQCRTMH          PR          EXTPROC('MQCRTMH')
D* Connection handle
D HCONN          10I 0 VALUE
D* Options that control the action of MQCRTMH
```

D CRTOPT	12A
D* Message handle	
D HMSG	20I 0
D* Completion code	
D CMPCOD	10I 0
D* Reason code qualifying CompCode	
D REASON	10I 0

## IBM i MQCTL (retorno de chamada de controle) em IBM i

A chamada MQCTL executa ações de controle nas manipulações de objetos abertas para uma conexão

- [“Sintaxe” na página 1331](#)
- [“Observações de Uso” na página 1331](#)
- [“Parâmetros” na página 1331](#)
- [“Declaração de RPG” na página 1336](#)

### Sintaxe

MQCTL (*Hconn, Operation, ControlOpts, CompCode, Reason*)

### Observações de Uso

1. As rotinas de retorno de chamada devem verificar as respostas de todos os serviços que elas chamam e se a rotina detectar uma condição que não pode ser resolvida, ela deverá emitir um comando MQCB (CBREG) para evitar chamadas repetidas para a rotina de retorno de chamada.

### Parâmetros

A chamada MQCTL possui os parâmetros a seguir:

#### HCONN (número inteiro assinado de 10 dígitos)-entrada

Esta manipulação representa a conexão ao gerenciador de filas. O valor de *HCONN* foi retornado por uma chamada anterior MQCONN ou MQCONNX.

#### OPERATN (número inteiro assinado de 10 dígitos)-entrada

A operação sendo processada no retorno de chamada definido para o identificador de objeto especificado. Deve-se especificar uma e apenas uma das opções a seguir:

#### CTLSR

Iniciar o consumo de mensagens para todas as funções de consumidor de mensagens definidas para o identificador de conexão especificado

Retornos de chamada executados em um encadeamento iniciado pelo sistema, que é diferente de qualquer um dos encadeamentos de aplicativos.

Esta operação fornece o controle do identificador de conexão fornecido para o sistema. As únicas chamadas MQI que podem ser emitidas por um encadeamento diferente do encadeamento do consumidor são:

- MQCTL com a Operação CTLSP
- MQCTL com Operação CTLSU
- MQDISC-Isso executa MQCTL com Operação CTLSP antes de desconectar o HConn.

RC2500 será retornado se uma chamada API IBM MQ for emitida enquanto a manipulação de conexões for iniciada e a chamada não for originada de uma função do consumidor de mensagens.

Se uma conexão falhar, isso interromperá a conversa o mais rápido possível. Portanto, é possível que uma chamada API IBM MQ que está sendo emitida no encadeamento principal receba o código de retorno RC2500 por um tempo, seguido pelo código de retorno RC2009 quando a conexão for revertida para o estado pausado.

Isso pode ser emitido em uma função de consumidor Para a mesma conexão que a rotina de retorno de chamada, seu único propósito é cancelar uma operação do CTLSW emitida anteriormente

Essa opção não será suportada se o aplicativo for ligado a uma biblioteca não encadeada do IBM MQ

### **CTLSW**

Iniciar o consumo de mensagens para todas as funções de consumidor de mensagens definidas para o identificador de conexão especificado

Os consumidores de mensagens são executados no mesmo encadeamento e o controle não é retornado ao responsável pela chamada de MQCTL até que:

- Liberado pelo uso das operações MQCTL CTLSW ou CTLSU ou
- Todas as rotinas do consumidor tiveram o registro removido ou suspenso.

Se todos os consumidores tiverem o registro removido ou suspenso, uma operação CTLSW implícita será emitida.

Esta opção não pode ser utilizada a partir de uma rotina de retorno de chamada, seja para o identificador de conexão atual ou qualquer outro identificador de conexão Se a chamada for tentada, será retornado com RC2012.

Se, a qualquer momento durante uma operação CTLSW, não houver consumidores não suspensos registrados, a chamada falhará com um código de razão de RC2446.

Se, durante uma operação CTLSW, a conexão for suspensa, a chamada MQCTL retornará um código de razão de aviso de RC2521; a conexão permanece 'iniciada'.

O aplicativo pode escolher emitir CTLSW ou CTLSU. Nesta instância, a operação CTLSU bloqueia.

Essa opção não é suportada em um cliente de encadeamento único..

### **CTLSW**

Pare o consumo de mensagens e aguarde todos os consumidores concluírem suas operações antes que essa opção seja concluída. Esta operação libera o identificador da conexão

Se emitido a partir de uma rotina de retorno de chamada, essa opção não entrará em vigor até que a rotina seja encerrada Mais nenhuma rotina do consumidor de mensagens é chamada depois que as rotinas do consumidor para mensagens já lidas foram concluídas e depois que as chamadas de parada (se solicitadas) para rotinas de retorno de chamada foram feitas.

Se emitido fora de uma rotina de retorno de chamada, o controle não retornará ao responsável pela chamada até que as rotinas do consumidor para mensagens já lidas tenham sido concluídas e após chamadas de parada (se solicitadas) para retornos de chamada terem sido feitas. Os retornos de chamada, no entanto, permanecem registrados.

Esta função não tem efeito sobre as mensagens de leitura antecipada. Deve-se assegurar que os consumidores executem MQCLOSE (COQSC), a partir da função de retorno de chamada, para determinar se há mensagens adicionais disponíveis para serem entregues

### **CTLSU**

Pausar o consumo de mensagens Esta operação libera o identificador da conexão

Isso não afeta a leitura à frente de mensagens para o aplicativo Se você pretende parar de consumir mensagens por um longo período, considere fechar a fila e reabri-la quando o consumo tiver que continuar

Se emitido a partir de uma rotina de retorno de chamada, ele não entrará em vigor até que a rotina seja encerrada Mais nenhuma rotina do consumidor de mensagens será chamada após a saída da rotina atual

Se emitido fora de um retorno de chamada, o controle não retornará ao responsável pela chamada até que a rotina do consumidor atual tenha sido concluída e nenhum outro seja chamado.



**CTLRE**

Retome o consumo de mensagens

Essa opção é normalmente emitida a partir do encadeamento do aplicativo principal, mas também pode ser usada a partir de uma rotina de retorno de chamada para cancelar um pedido de suspensão anterior emitido na mesma rotina

Se CTLRE for usado para continuar um CTLSW, então a operação será bloqueada.

**PCTLOP (MQCTLO)-entrada**

Opções que controlam a ação de MQCTL

Consulte [MQCTLO](#) para obter detalhes da estrutura..

**CMPCOD (número inteiro assinado de 10 dígitos)-saída**

O código de conclusão; é um dos seguintes:

**CCOK**

Indica conclusão bem-sucedida.

**CCWARN**

Aviso (conclusão parcial).

**CCFAIL**

A chamada falhou.

**REASON (número inteiro assinado de 10 dígitos)-saída**

Os seguintes códigos de razão são aqueles que o gerenciador de filas pode retornar para o parâmetro **Reason** .

Se *CMPCOD* for CCOK:

**RCNONE**

(0, X'000') Nenhuma razão a relatar.

Se *CMPCOD* for CCFAIL:

**RC2133**

(2133, X'855') Impossível carregar módulos de serviços de conversão de dados.

**RC2204**

(2204, X'89C') Adaptador não disponível.

**RC2130**

(2130, X'852') Não foi possível carregar o módulo de serviço do adaptador.

**RC2374**

(2374, X'946') Falha na saída de API.

**RC2183**

(2183, X'887') Não foi possível carregar a saída de API.

**RC2157**

(2157, X'86D') ASIDs de início e primárias são diferentes.

**RC2005**

(2005, X'7D5') Parâmetro de comprimento de buffer inválido.

**RC2487**

(2487, X'9B7') Não é possível chamar a rotina de retorno de chamada

**RC2448**

(2448, X' 990 ') Não é possível remover o registro, suspender ou continuar porque não há retorno de chamada registrado

**RC2486**

(2486, X'9B6') Ou CallbackFunction e CallbackName foram especificados em uma chamada CBREG ou um de CallbackFunction ou CallbackName foi especificado, mas não corresponde à função de retorno de chamada atualmente registrada.

- RC2483**  
(2483, X'9B3') Campo de tipo Callbackincorreto
- RC2219**  
(2219, X'8AB') Chamada MQI inserida antes da chamada anterior concluída.
- RC2444**  
(2444, X'98C') O bloco de opções está incorreto
- RC2484**  
(2484, X'9B4') Campo de opções MQCBD incorreto.
- RC2140**  
(2140, X'85C') Pedido de espera rejeitado por CICS.
- RC2009**  
(2009, X'7D9') Conexão ao gerenciador de filas perdida.
- RC2217**  
(2217, X'8A9') Não autorizado para conexão.
- RC2202**  
(2202, X'89A') Quiesce de conexão.
- RC2203**  
(2203, X'89B') Conexão sendo encerrada.
- RC2207**  
(2207, X'89F') Erro de identificador de correlação.
- RC2016**  
(2016, X'7E0 ') É inibida para a fila.
- RC2351**  
(2351, X'92F') Unidades de trabalho globais conflitam.
- RC2186**  
(2186, X'88A') Estrutura de opções de obtenção de mensagem inválida.
- RC2353**  
(2353, X'931') Manipulação em uso para unidade de trabalho global.
- RC2018**  
(2018, X'7E2') Manipulação de conexões não válida.
- RC2019**  
(2019, X'7E3') Manipulação de objetos não válida.
- RC2259**  
(2259, X'8D3') Especificação de pesquisa inconsistente.
- RC2245**  
(2245, X'8C5') Especificação de unidade de trabalho inconsistente.
- RC2246**  
(2246, X'8C6 ') Mensagem sob o cursor inválida para recuperação.
- RC2352**  
(2352, X'930') Unidade de trabalho global conflita com unidade de trabalho local.
- RC2247**  
(2247, X'8C7') Opções correspondentes inválidas.
- RC2485**  
(2485, X'9B5') Campo de comprimento MaxMsgincorreto
- RC2026**  
(2026, X'7EA') Descritor de mensagem inválido.
- RC2497**  
(2497, X'9C1') O ponto de entrada de função especificado não foi localizado no módulo

**RC2496**

(2496, X'9C0') O módulo foi localizado, mas é do tipo errado (32 bits ou 64 bits) ou não é um dll válido.

**RC2495**

(2495, X'9BF') Módulo não localizado no caminho da procura ou não autorizado a carregar.

**RC2206**

(2206, X'89E') Erro de identificador de mensagem.

**RC2250**

(2250, X'8CA') Número de sequência da mensagem inválido.

**RC2331**

(2331, X'91B') Uso de token da mensagem inválido.

**RC2036**

(2036, X'7F4') Fila não aberta para navegação.

**RC2037**

(2037, X'7F5') Fila não aberta para entrada.

**RC2041**

(2041, X'7F9') Definição de objeto alterada desde a abertura.

**RC2101**

(2101, X'835') Objeto danificado.

**RC2488**

(2488, X'9B8') Código de operação incorreto na chamada API

**RC2046**

(2046, X'7FE') Opções não válidas ou não consistentes.

**RC2193**

(2193, X'891') Erro ao acessar o conjunto de dados do conjunto de páginas.

**RC2052**

(2052, X'804') A fila foi excluída.

**RC2394**

(2394, X'95A') Fila possui tipo de índice errado.

**RC2058**

(2058, X'80A') Nome do gerenciador de filas não válido ou não conhecido.

**RC2059**

(2059, X'80B') Gerenciador de filas não disponível para conexão.

**RC2161**

(2161, X'871') Quiesce do gerenciador de filas.

**RC2162**

(2162, X'872') Gerenciador de filas sendo encerrado.

**RC2102**

(2102, X'836') Recursos insuficientes do sistema disponíveis.

**RC2069**

(2069, X'815') Sinais pendentes para essa manipulação.

**RC2071**

(2071, X'817') Armazenamento insuficiente disponível.

**RC2109**

(2109, X'83D') Chamada suprimida pelo programa de saída.

**RC2072**

(2072, X'818 ') Suporte de ponto de sincronização não disponível.

**RC2195**

(2195, X'893') Ocorreu um erro inesperado.

**RC2354**

(2354, X'932') Cadastramento na unidade de trabalho global falhou.

**RC2355**

(2355, X'933') Combinação de chamadas de unidade de trabalho não suportada.

**RC2255**

(2255, X'8CF') Unidade de trabalho não disponível para o gerenciador de filas a usar.

**RC2090**

(2090, X'82A') Intervalo de Espera em MQGMO inválido.

**RC2256**

(2256, X'8D0') Versão errada do MQGMO fornecido.

**RC2257**

(2257, X'8D1') Versão errada do MQMD fornecido.

**RC2298**

(2298, X'8FA') A função solicitada não está disponível no ambiente atual.

**Declaração de RPG**

```

C*.1.....2.....3.....4.....5.....6.....7..
C                                CALLP      MQCTL(HCONN : OPERATN : PCTLOP :
                                CMPCOD : REASON)

```

A definição de protótipo para a chamada é:

```

DMQCTL          PR          EXTPROC('MQCTL')
D* Connection handle
D HCONN          10I 0 VALUE
D* Operation
D OPERATN       10I 0 VALUE
D* Control options
D PCTLOP        32A
D* Completion code
D CMPCOD        10I 0
D* Reason code qualifying CompCode
D REASON        10I 0

```

**IBM i MQDISC (Desconectar o gerenciador de filas) no IBM i**

A chamada MQDISC quebra a conexão entre o gerenciador de fila e o programa de aplicativo e é o inverso da chamada MQCONN ou MQCONNX.

- [“Sintaxe” na página 1336](#)
- [“Observações de Uso” na página 1336](#)
- [“Parâmetros” na página 1337](#)
- [“Declaração de RPG” na página 1338](#)

**Sintaxe**

MQDISC (HCONN, CMPCOD, REASON)

**Observações de Uso**

1. Se uma chamada MQDISC for emitida quando o aplicativo ainda tiver objetos abertos, esses objetos serão encerrados pelo gerenciador de filas, com as opções de fechamento configuradas como CONONE

2. Se o aplicativo terminar com mudanças não confirmadas em uma unidade de trabalho, a disposição dessas mudanças dependerá de como o aplicativo termina:
  - a. Se o aplicativo emitir a chamada MQDISC antes de terminar:
    - Para uma unidade de trabalho coordenada do gerenciador de filas, o gerenciador de filas emite a chamada MQCMIT em nome do aplicativo. A unidade de trabalho é confirmada, se possível, e restaurada, se não.
    - Para uma unidade de trabalho coordenada externamente, não há nenhuma alteração no status da unidade de serviço; no entanto, o gerenciador de filas indicará que a unidade de trabalho deve ser confirmada, quando solicitado pelo coordenador da unidade de trabalho
  - b. Se o aplicativo terminar normalmente, mas sem emitir a chamada MQDISC, a unidade de trabalho será restaurada.
  - c. Se o aplicativo terminar *de forma anormal* sem a emissão da chamada MQDISC, a unidade de trabalho será restaurada para fora.

## Parâmetros

A chamada MQDISC possui os parâmetros a seguir:

### **HCONN (número inteiro assinado de 10 dígitos)-entrada/saída**

Manipulação de conexões.

Esta manipulação representa a conexão ao gerenciador de filas. O valor de *HCONN* foi retornado por uma chamada anterior MQCONN ou MQCONNX.

Na conclusão bem-sucedida da chamada, o gerenciador de filas configura *HCONN* para um valor que não é um identificador válido do ambiente. Esse valor é:

#### **HCUNUH**

Identificador de conexão não utilizável

### **CMPCOD (número inteiro assinado de 10 dígitos)-saída**

Código de conclusão.

É um dos seguintes:

#### **CCOK**

Indica conclusão bem-sucedida.

#### **CCWARN**

Aviso (conclusão parcial).

#### **CCFAIL**

A chamada falhou.

### **REASON (número inteiro assinado de 10 dígitos)-saída**

Código de razão qualificando *CMPCOD*.

Se *CMPCOD* for CCOK:

#### **RCNONE**

(0, X'000') Nenhuma razão a relatar.

Se *CMPCOD* for CCFAIL:

#### **RC2219**

(2219, X'8AB') chamada MQI reinserida antes da chamada anterior ser concluída.

#### **RC2009**

(2009, X'7D9') Conexão ao gerenciador de filas perdida.

#### **RC2018**

(2018, X'7E2') Manipulação de conexões não válida.

**RC2058**

(2058, X'80A') Nome do gerenciador de filas não válido ou não conhecido.

**RC2059**

(2059, X'80B') Gerenciador de filas não disponível para conexão.

**RC2162**

(2162, X'872') Gerenciador de filas sendo encerrado.

**RC2102**

(2102, X'836') Recursos insuficientes do sistema disponíveis.

**RC2071**

(2071, X'817') Armazenamento insuficiente disponível.

**RC2195**

(2195, X'893') Ocorreu um erro inesperado.

**Declaração de RPG**

```

C*..1.....2.....3.....4.....5.....6.....7..
C                               CALLP      MQDISC(HCONN : CMPCOD : REASON)

```

A definição de protótipo para a chamada é:

```

D*..1.....2.....3.....4.....5.....6.....7..
DMQDISC          PR          EXTPROC('MQDISC')
D* Connection handle
D HCONN          10I 0
D* Completion code
D CMPCOD         10I 0
D* Reason code qualifying CMPCOD
D REASON         10I 0

```

**IBM i MQDLTMH (Excluir identificador de mensagens) no IBM i**

A chamada MQDLTMH exclui um identificador de mensagens e é o inverso da chamada MQCRTMH

- [“Sintaxe” na página 1338](#)
- [“Observações de Uso” na página 1338](#)
- [“Parâmetros” na página 1340](#)
- [“Declaração de RPG” na página 1341](#)

**Sintaxe**

MQDLTMH ((*Hconn*, *Hmsg*, *DltMsgHOpts*, *CompCode*, *Reason*))

**Observações de Uso**

1. É possível usar essa chamada apenas quando o próprio gerenciador de filas coordena a unidade de trabalho. Ele pode ser:
  - Uma unidade de trabalho local, em que as mudanças afetam apenas os recursos do IBM MQ
  - Uma unidade de trabalho global, na qual as mudanças podem afetar recursos pertencentes a outros gerenciadores de recursos, bem como afetar recursos do IBM MQ.

Para obter detalhes adicionais sobre as unidades de trabalho locais e globais, consulte [“MQBEGIN \(Iniciar unidade de trabalho\) em IBM i” na página 1299](#)
2. Em ambientes nos quais o gerenciador de filas não coordena a unidade de trabalho, use a chamada de retorno apropriada, em vez de MQBACK. O ambiente também pode suportar um retorno implícito causado pela finalização anormal do aplicativo.

- No z/OS, use as seguintes chamadas:
    - Programas em lote (incluindo IMS programas DL/I em lote) podem usar a chamada MQBACK se a unidade de trabalho afetar apenas recursos IBM MQ . No entanto, se a unidade de trabalho afetar recursos do IBM MQ e recursos pertencentes a outros gerenciadores de recursos (por exemplo, Db2 ), use a chamada SRRBACK fornecida pelo z/OS Recoverable Resource Service (RRS). A chamada SRRBACK retorna mudanças em recursos pertencentes aos gerenciadores de recursos que foram ativados para coordenação RRS.
    - Aplicativos CICS devem usar o comando EXEC CICS SYNCPOINT ROLLBACK para voltar a unidade de trabalho. Não use a chamada MQBACK para os aplicativos CICS
    - Os aplicativos IMS (que não sejam programas DL/I em lote) devem usar chamadas IMS como ROLB para voltar a unidade de trabalho. Não use a chamada MQBACK para aplicativos IMS (exceto programas DL/I em lote).
  - No IBM i, use essa chamada para unidades locais de trabalho coordenadas pelo gerenciador de fila.. Isso significa que uma definição de confirmação não deve existir no nível da tarefa, ou seja, o comando STRCMTCTL com o parâmetro **CMTSCOPE (\*JOB)** não deve ter sido emitido para a tarefa
3. Se um aplicativo terminar com mudanças não confirmados em uma unidade de trabalho, a disposição dessas mudanças dependerá se o aplicativo será encerrado normalmente ou de forma anormal. Consulte as notas de uso em [“MQDISC \(Desconectar o gerenciador de filas\) no IBM i” na página 1336](#) para obter detalhes adicionais
4. Quando um aplicativo coloca ou obtém mensagens em grupos ou segmentos de mensagens lógicas, o gerenciador de filas retém informações relacionadas ao grupo de mensagens e mensagens lógicas para as últimas chamadas MQPUT e MQGET bem-sucedidas. Essas informações estão associadas ao identificador de filas e incluem itens como:
- Os valores dos campos *GroupId*, *MsgSeqNumber*, *Offset* e *MsgFlags* em MQMD..
  - Indica se a mensagem faz parte de uma unidade de trabalho
  - Para a chamada MQPUT: se a mensagem é persistente ou não persistente.

O gerenciador de filas mantém três conjuntos de informações de grupo e segmento, um conjunto para cada um dos seguintes:

- A última chamada MQPUT bem-sucedida (isso pode fazer parte de uma unidade de trabalho)
- A última chamada MQGET bem-sucedida que removeu uma mensagem da fila (isso pode fazer parte de uma unidade de trabalho).
- A última chamada MQGET bem-sucedida que navegou em uma mensagem na fila (isso não pode fazer parte de uma unidade de trabalho)

Se o aplicativo colocar ou receber as mensagens como parte de uma unidade de trabalho e o aplicativo, em seguida, restaurar a unidade de trabalho, as informações do grupo e do segmento serão restauradas para o valor que ele tinha anteriormente:

- As informações associadas à chamada MQPUT são restauradas para o valor que ela tinha antes da primeira chamada MQPUT bem-sucedida para essa manipulação de fila na unidade de trabalho atual..
- As informações associadas à chamada MQGET são restauradas para o valor que ela tinha antes da primeira chamada MQGET bem-sucedida para essa manipulação de fila na unidade de trabalho atual...

As filas que foram atualizadas pelo aplicativo após a unidade de trabalho ter iniciado, mas fora do escopo da unidade de trabalho, não terão suas informações de grupo e de segmento restauradas se a unidade de trabalho for restaurada.

A restauração das informações do grupo e do segmento para seu valor anterior quando uma unidade de trabalho é restaurada permite que o aplicativo difunda um grupo de mensagens grandes ou uma mensagem lógica grande que consiste em muitos segmentos em várias unidades de trabalho e reinicie no ponto correto no grupo de mensagens ou mensagem lógica se uma das unidades de trabalho falhar. Usar várias unidades de trabalho pode ser vantajoso se o gerenciador de fila local tiver apenas

armazenamento de fila limitado. No entanto, o aplicativo deve manter informações suficientes para poder reiniciar colocando ou obtendo mensagens no ponto correto se ocorrer uma falha do sistema.

Para obter detalhes de como reiniciar no ponto correto após uma falha do sistema, consulte a opção **PMLOGO** descrita em **PMOPT (número inteiro assinado de 10 dígitos)** e a opção **GMLOGO** descrita em **GMOPT (número inteiro assinado de 10 dígitos)**.

As notas de uso restantes se aplicam apenas quando o gerenciador de filas coordena as unidades de trabalho:

5. Uma unidade de trabalho tem o mesmo escopo que uma manipulação de conexões.. Todas as chamadas de IBM MQ que afetam uma unidade de trabalho específica devem ser executadas usando a mesma manipulação de conexões Chamadas emitidas usando um identificador de conexão diferente (por exemplo, chamadas emitidas por outro aplicativo) afetam uma unidade de trabalho diferente. Consulte **HCONN (número inteiro assinado de 10 dígitos)-saída** para obter informações sobre o escopo de identificadores de conexões...
6. Apenas as mensagens que foram colocadas ou recuperadas como parte da unidade de trabalho atual são afetada por esta chamada
7. Um aplicativo de longa execução que emite chamadas **MQGET** ou **MQPUT** ou **MQPUT1** dentro de uma unidade de trabalho, mas que nunca emite uma confirmação ou chamada de restauração, pode preencher filas com mensagens que não estão disponíveis para outros aplicativos. Para proteger contra essa possibilidade, o administrador deve configurar o atributo do gerenciador de filas **MaxUncommittedMsgs** para um valor que seja baixo o suficiente para evitar que os aplicativos runaway preencham as filas, mas alto o suficiente para permitir que os aplicativos do sistema de mensagens esperados funcionem corretamente.

## Parâmetros

A chamada **MQDLTMH** possui os parâmetros a seguir:

### **HCONN (número inteiro assinado de 10 dígitos)-entrada**

Esta manipulação representa a conexão ao gerenciador de filas.

O valor deve corresponder ao identificador de conexão que foi usado para criar o identificador de mensagem especificado no parâmetro **HMSG** ...

Se o identificador de mensagem foi criado usando **HCUNAS** então uma conexão válida deve ser estabelecida no encadeamento excluindo o identificador de mensagem, caso contrário, a chamada falhará com **RC2009**

### **HMSG (número inteiro assinado de 20 dígitos)-entrada/saída**

Esta é a manipulação de mensagem a ser excluída O valor foi retornado por uma chamada **MQCRTMH** anterior.

Na conclusão bem-sucedida da chamada, o identificador é configurado para um valor inválido para o ambiente. Esse valor é:.

#### **HMUNUH**

Identificador de mensagem não utilizável

O identificador de mensagem não poderá ser excluído se outra chamada IBM MQ estiver em andamento e tiver passado o mesmo identificador de mensagem...

### **DLTOPT (MQDMHO)-entrada**

Consulte **MQDMHO** para obter detalhes..

### **CMPCOD (número inteiro assinado de 10 dígitos)-saída**

O código de conclusão; é um dos seguintes:

#### **CCOK**

Indica conclusão bem-sucedida.



**CCFAIL**

A chamada falhou.

**REASON (número inteiro assinado de 10 dígitos)-saída**

O código de razão qualificando *CMPCOD*.

Se *CMPCOD* for CCOK:

**RCNONE**

(0, X'000') Nenhuma razão a relatar.

Se *CMPCOD* for CCFAIL:

**RC2204**

(2204, X'089C') Adaptador não disponível.

**RC2130**

(2130, X'852') Não foi possível carregar o módulo de serviço do adaptador.

**RC2157**

(2157, X'86D') ASIDs de início e primárias são diferentes.

**RC2219**

(2219, X'08AB') chamada MQI inserida antes da chamada anterior ser concluída.

**RC2009**

(2009, X'07D9') Conexão com o gerenciador de filas perdida.

**RC2462**

(2462, X'099E') A estrutura de opções de manipulação de mensagens de exclusão não é válida...

**RC2460**

(2460, X'099C') Ponteiro do identificador de mensagens inválido.

**RC2499**

(2499, X'09C3') Identificador de mensagem já em uso.

**RC2046**

(2046, X'07FE') Opções não válidas ou não consistentes.

**RC2071**

(2071, X'817') Armazenamento insuficiente disponível.

**RC2195**

(2195, X'893') Ocorreu um erro inesperado.

Consulte [“Códigos de retorno para IBM i \(ILE RPG\)”](#) na página 1477 para obter mais detalhes.

**Declaração de RPG**

```
C*.1.....2.....3.....4.....5.....6.....7..
C          CALLP      MQDLTMH(HCONN : HMSG : DLTOPT :
                   CMPCOD : REASON)
```

A definição de protótipo para a chamada é:

```
MQDLTMH          PR          EXTPROC('MQDLTMH')
D* Connection handle
D HCONN          10I 0 VALUE
D* Message handle
D HMSG          20I 0
D* Options that control the action of MQDLTMH
D DLTOPT        12A
D* Completion code
D CMPCOD        10I 0
D* Reason code qualifying CompCode
D REASON        10I 0
```

## **MQDLTMP-Excluir propriedade de mensagem**

A chamada MQDLTMP exclui uma propriedade de um identificador de mensagem e é o inverso da chamada MQSETMP

- “[Sintaxe](#)” na página 1342
- “[Parâmetros](#)” na página 1342
- “[Declaração de RPG](#)” na página 1343

### **Sintaxe**

MQDLTMP (*Hconn, Hmsg, DltPropOpts, Name, CompCode, Reason*)

### **Parâmetros**

A chamada MQDLTMP possui os parâmetros a seguir:

#### **HCONN (número inteiro assinado de 10 dígitos)-Entrada**

Esta manipulação representa a conexão ao gerenciador de filas. O valor deve corresponder ao identificador de conexão que foi usado para criar o identificador de mensagem especificado no parâmetro **HMSG** ...

Se o identificador de mensagem foi criado usando HCUNAS, uma conexão válida deverá ser estabelecida no encadeamento excluindo o identificador de mensagem, caso contrário, a chamada falhará com RC2009

#### **HMSG (número inteiro assinado de 20 dígitos)-entrada**

Esta é a manipulação de mensagem que contém a propriedade a ser excluída. O valor foi retornado por uma chamada MQCRTMH anterior.

#### **DLTOPT (MQDMPO)-Entrada**

Consulte o tipo de dados [MQDMPO](#) para obter detalhes..

#### **PRNAME (MQCHARV)-entrada**

O nome da propriedade a excluir. Consulte [Nomes de Propriedades](#) , para obter informações adicionais sobre nomes de propriedade

Os curingas não são permitidos no nome da propriedade

#### **CMPCOD (número inteiro assinado de 10 dígitos)-saída**

O código de conclusão; é um dos seguintes:

##### **CCOK**

Indica conclusão bem-sucedida.

##### **CCWARN**

Aviso (conclusão parcial).

##### **CCFAIL**

A chamada falhou.

#### **REASON (número inteiro assinado de 10 dígitos)-saída**

O código de razão qualificando *CMPCOD*.

Se *CMPCOD* for CCOK:

##### **RCNONE**

(0, X'000') Nenhuma razão a relatar.

Se *CMPCOD* for CCWARN:

##### **RC2471**

(2471, X'09A7') Propriedade não disponível.

**RC2421**

(2421, X'0975 ') Uma pasta MQRFH2 contendo propriedades não pôde ser analisada.

Se *CMPCOD* for CCFAIL:

**RC2204**

(2204, X'089C') Adaptador não disponível.

**RC2130**

(2130, X'0852 ') Impossível carregar o módulo de serviço do adaptador.

**RC2157**

(2157, X'086D') Os ASIDs primário e inicial diferem.

**RC2219**

(2219, X'08AB') chamada MQI inserida antes da chamada anterior ser concluída.

**RC2009**

(2009, X'07D9') Conexão com o gerenciador de filas perdida.

**RC2481**

(2481, X'09B1') A estrutura de opções de propriedade de mensagem de exclusão não é válida..

**RC2460**

(2460, X'099C') Identificador de mensagem inválido.

**RC2499**

(2499, X'09C3') Identificador de mensagem já em uso.

**RC2046**

(2046, X'07FE') Opções não válidas ou não consistentes.

**RC2442**

(2442, X'098A') Nome da propriedade inválido

**RC2111**

(2111, X'083F') Identificador do conjunto de caracteres codificados do nome da propriedade inválido.

**RC2195**

(2195, X'0893 ') Ocorreu um erro inesperado.

Para obter mais informações sobre esses códigos, consulte [Códigos de conclusão e de razão da API](#)

**Declaração de RPG**

```

C*.1.....2.....3.....4.....5.....6.....7..
C          CALLP      MQDLTMP(HCONN : HMSG : DLTOPT :
                        PRNAME : CMPCOD : REASON)

```

A definição de protótipo para a chamada é:

```

DMQDLTMP      PR          EXTPROC('MQDLTMP')
D* Connection handle
D HCONN          10I 0 VALUE
D* Message handle
D HMSG          20I 0 VALUE
D* Option of MQDLTMP
D DLTOPT          12A
D* Property name
D PRNAME          32A
D* Completion code
D CMPCOD          10I 0
D* Reason code qualifying CompCode
D REASON          10I 0

```

A chamada MQGET recupera uma mensagem de uma fila local que foi aberta usando a chamada MQOPEN.

- [“Sintaxe” na página 1344](#)
- [“Observações de Uso” na página 1344](#)
- [“Parâmetros” na página 1347](#)
- [“Declaração de RPG” na página 1352](#)

## Sintaxe

MQGET (*HCONN, HOBJ, MSGDSC, GMO, BUFLN, BUFFER, DATLEN, CMPCOD, REASON*)

## Observações de Uso

1. A mensagem recuperada normalmente é excluída da fila. Essa exclusão pode ocorrer como parte da própria chamada MQGET ou como parte de um ponto de sincronização. A exclusão de mensagem não ocorre se uma opção GMBRWF ou GMBRWN for especificada no parâmetro **GMO** (consulte o campo *GMOPT* descrito em [“MQGMO \(opções Get-message\) em IBM i” na página 1113](#) ).
2. Se a opção GMLK for especificada com uma das opções de navegação, a mensagem procurada será bloqueada para que ela fique visível apenas para essa manipulação

Se a opção GMUNLK for especificada, uma mensagem bloqueada anteriormente será desbloqueada. Nenhuma mensagem é recuperada nesse caso e os parâmetros **MSGDSC, BUFLN, BUFFER e DATLEN** não são verificados ou alterados.

3. Se o aplicativo emitindo a chamada MQGET estiver em execução como um IBM MQ MQI client, será possível que a mensagem recuperada seja perdida se durante o processamento da chamada MQGET o IBM MQ MQI client for finalizado de forma anormal ou a conexão do cliente for interrompida. Isso ocorre porque o substituto que está em execução na plataforma do gerenciador de filas e que emite a chamada MQGET em nome do cliente não pode detectar a perda do cliente até que o substituto esteja prestes a retornar a mensagem para o cliente; isso ocorre após a mensagem ter sido removida da fila. Isso pode ocorrer para mensagens persistentes e mensagens não persistentes.

O risco de perder mensagens dessa maneira pode ser eliminado sempre recuperando mensagens dentro de unidades de trabalho (ou seja, especificando a opção *GMSYP* na chamada MQGET e usando as chamadas *MQCMIT* ou *MQBACK* para confirmar ou voltar a unidade de trabalho quando o processamento da mensagem estiver concluído). Se *GMSYP* for especificado e o cliente terminar de forma anormal ou a conexão for interrompida, o substituto restaurará a unidade de trabalho no gerenciador de filas e a mensagem será restabelecida na fila.

Em princípio, a mesma situação pode surgir com aplicativos que estão em execução na plataforma do gerenciador de fila, mas nesse caso a janela durante a qual uma mensagem pode ser perdida é pequena. No entanto, como com IBM MQ MQI clients, o risco pode ser eliminado recuperando a mensagem em uma unidade de trabalho.

4. Se um aplicativo colocar uma sequência de mensagens em uma fila específica em uma única unidade de trabalho e, em seguida, confirmar essa unidade de trabalho com êxito, as mensagens ficarão disponíveis para recuperação conforme a seguir:
  - Se a fila for uma *fila não compartilhada* (ou seja, uma fila local), todas as mensagens na unidade de trabalho se tornarão disponíveis ao mesmo tempo.
  - Se a fila for uma *fila compartilhada*, as mensagens na unidade de trabalho se tornarão disponíveis na ordem em que foram colocadas, mas não todas ao mesmo tempo. Quando o sistema está pesadamente carregado, é possível que a primeira mensagem na unidade de trabalho seja recuperada com êxito, mas para a chamada MQGET para a segunda mensagem ou mensagem

subsequente na unidade de trabalho falhar com RC2033. Se isso ocorrer, o aplicativo deverá aguardar um curto tempo e, em seguida, tentar a operação novamente

5. Se um aplicativo colocar uma sequência de mensagens na mesma fila sem usar grupos de mensagens, a ordem dessas mensagens será preservada desde que determinadas condições sejam satisfeitas. Consulte as notas de uso na descrição da chamada MQPUT para obter detalhes. Se as condições forem satisfeitas, as mensagens serão apresentadas no aplicativo de recebimento na ordem na qual elas foram enviadas, desde que:

- Somente um receptor esteja recebendo mensagens da fila.

Se houver dois ou mais aplicativos recebendo mensagens da fila, eles deverão concordar com o emissor que o mecanismo seja usado para identificar mensagens que pertencem a uma sequência. Por exemplo, o emissor pode configurar todos os campos MDCID nas mensagens em uma sequência para um valor que era exclusivo para essa sequência de mensagens...

- O receptor não altera deliberadamente a ordem de recuperação, por exemplo, especificando um determinado MDMID ou MDCID

Se o aplicativo de envio colocar as mensagens como um grupo de mensagens, as mensagens serão apresentadas ao aplicativo de recebimento na ordem correta se o aplicativo de recebimento especificar a opção GMLOGO na chamada MQGET.. Para obter informações adicionais sobre grupos de mensagens, consulte:

- MDMFL Campo no MQMD
- Opção PMLOGO em MQPMO
- Opção GMLOGO em MQGMO

6. Aplicativos testam o código de feedback FBQUIT no campo MDFB do parâmetro **MSGDSC** . Se esse valor for localizado, o aplicativo será encerrado. Consulte o campo MDFB descrito em “MQMD (Descritor de mensagens) em IBM i” na página 1148 , para obter mais informações

7. Se a fila identificada por HOBJ foi aberta com a opção OOSAVA e o código de conclusão da chamada MQGET for CCOK ou CCWARN, o contexto associado ao identificador de filas HOBJ será configurado para o contexto da mensagem que foi recuperada (a menos que a opção GMBRWF ou GMBRWN seja configurada; nesse caso, o contexto será marcado como não disponível)... Esse contexto pode ser usado em uma chamada MQPUT ou MQPUT1 subsequente, especificando as opções PMPASI ou PMPASA. Isto permite que o contexto da mensagem recebida seja transferido no todo ou em parte para outra mensagem (por exemplo, quando a mensagem é encaminhada para outra fila). Para obter mais informações sobre o contexto de mensagem, consulte [Contexto da mensagem](#) e [Informações de contexto de controle](#).

8. Se a opção GMCONV for incluída no parâmetro **GMO** , os dados da mensagem do aplicativo serão convertidos na representação solicitada pelo aplicativo de recebimento, antes que os dados sejam colocados no parâmetro **BUFFER** :

- O campo MDFMT nas informações de controle na mensagem identifica a estrutura de dados do aplicativo e os campos MDCSI e MDENC nas informações de controle na mensagem especificam seu identificador e codificação do conjunto de caracteres.
- O aplicativo que emite a chamada MQGET especifica nos campos MDCSI e MDENC no parâmetro **MSGDSC** o identificador do conjunto de caracteres e a codificação para os quais os dados da mensagem do aplicativo devem ser convertidos..

Quando a conversão dos dados da mensagem for necessária, a conversão será executada pelo próprio gerenciador de fila ou por uma saída gravada pelo usuário, dependendo do valor do campo MDFMT nas informações de controle na mensagem:

- Os seguintes formatos são convertidos automaticamente pelo gerenciador de filas; esses formatos são chamados de formatos "integrados":

FMADMN

FMMDE

DECICS

FMPCF

FMCMD1	FMRMH
FMCMD2	FMRFH
FMDLH	FMRFH2
FMDH	FMSTR
FMEVNT	FMTM
DEIMS	FMXQH
FMIMVS	

- O nome do formato FMNONE é um valor especial que indica que a natureza dos dados na mensagem é indefinida.. Como consequência, o gerenciador de filas não tenta a conversão quando a mensagem é recuperada da fila.

**Nota:** Se GMCONV for especificado na chamada MQGET para uma mensagem que tenha um nome de formato FMNONE e o conjunto de caracteres ou a codificação da mensagem for diferente daquele especificado no parâmetro **MSGDSC** , a mensagem ainda será retornada no parâmetro **BUFFER** (assumindo que não haja outros erros), mas a chamada será concluída com o código de conclusão CCWARN e o código de razão RC2110.

FMNONE pode ser usado quando a natureza dos dados da mensagem significa que ela não requer conversão ou quando os aplicativos de envio e recebimento concordaram entre si o formulário no qual os dados da mensagem devem ser enviados

- Todos os outros nomes de formato fazem com que a mensagem seja transmitida para uma saída gravada pelo usuário para conversão A saída possui o mesmo nome que o formato, à parte das adições específicas do ambiente. Nomes de formato especificados pelo usuário não devem começar com as letras "MQ", pois tais nomes podem entrar em conflito com nomes de formato suportados no futuro.

Os dados do usuário na mensagem podem ser convertidos entre quaisquer conjuntos de caracteres e codificações suportados. No entanto, esteja ciente de que, se a mensagem contiver uma ou mais estruturas de cabeçalho IBM MQ , a mensagem não poderá ser convertida de ou para um conjunto de caracteres que possui caracteres de byte duplo ou multibyte para qualquer um dos caracteres que são válidos em nomes de filas. O código de razão RC2111 ou RC2115 resultará se isso for tentado e a mensagem não for convertida. O conjunto de caracteres Unicode UTF-16 é um exemplo desse conjunto de caracteres.

No retorno de MQGET, o código de razão a seguir indica que a mensagem foi convertida com êxito:

- RCNONE

O código de razão a seguir indica que a mensagem pode ter sido convertida com êxito; o aplicativo deve verificar os campos MDCSI e MDENC no parâmetro **MSGDSC** para descobrir:

- RC2079

Todos os outros códigos de razão indicam que a mensagem não foi convertida.

**Nota:** A interpretação do código de razão descrito neste exemplo é verdadeira para conversões executadas por saídas gravadas pelo usuário somente se a saída estiver em conformidade com as diretrizes de processamento

9. Para os formatos integrados listados anteriormente, o gerenciador de filas pode executar a conversão padrão de sequências de caracteres na mensagem quando a opção GMCONV for especificada A conversão padrão permite que o gerenciador de filas use um conjunto de caracteres padrão especificado pela instalação que aproxima o conjunto de caracteres real, ao converter dados de cadeia. Como resultado, a chamada MQGET pode ser bem-sucedida com o código de conclusão CCOK, em vez de concluir com CCWARN e o código de razão RC2111 ou RC2115.

**Nota:** O resultado de usar um conjunto de caracteres aproximado para converter dados de cadeia é que alguns caracteres podem ser convertidos incorretamente. Isso pode ser evitado usando na

sequência apenas caracteres que são comuns ao conjunto de caracteres real e ao conjunto de caracteres padrão.

A conversão padrão se aplica aos dados da mensagem do aplicativo e aos campos de caractere nas estruturas MQMD e MQMDE:

- A conversão padrão dos dados da mensagem do aplicativo ocorre apenas quando todas as seguintes instruções são verdadeiras:
  - O aplicativo especifica o GMCONV.
  - A mensagem contém dados que devem ser convertidos de ou para um conjunto de caracteres que não é suportado.
  - A conversão padrão foi ativada quando o gerenciador de filas foi instalado ou reiniciado.
- A conversão padrão dos campos de caractere nas estruturas MQMD e MQMDE ocorre conforme necessário, desde que a conversão padrão esteja ativada para o gerenciador de filas. A conversão é executada mesmo se a opção GMCONV não for especificada pelo aplicativo na chamada MQGET..

10. O parâmetro **BUFFER** mostrado no exemplo de programação de RPG é declarado como uma cadeia; isso restringe o comprimento máximo do parâmetro a 256 bytes. Se um buffer maior for necessário, o parâmetro deverá ser declarado como uma estrutura ou como um campo em um arquivo físico.

Declarar o parâmetro como uma estrutura aumenta o comprimento máximo possível para 9999 bytes, enquanto declarar o parâmetro como um campo em um arquivo físico aumenta o comprimento máximo possível para aproximadamente 32 KB.

## Parâmetros

A chamada MQGET tem os seguintes parâmetros:

### **HCONN (número inteiro assinado de 10 dígitos)-entrada**

Manipulação de conexões.

Esta manipulação representa a conexão ao gerenciador de filas. O valor de HCONN foi retornado por uma chamada anterior MQCONN ou MQCONNX.

### **HOBJ (número inteiro assinado de 10 dígitos)-entrada**

Manipulação de objetos.

Esse manipulador representa a fila da qual uma mensagem será recuperada. O valor de HOBJ foi retornado por uma chamada MQOPEN anterior. A fila deve ter sido aberta com uma ou mais das seguintes opções (consulte [“MQOPEN \(objeto aberto\) no IBM i”](#) na página 1369 para obter detalhes):

- OOINPS
- OOINPX
- OOINPQ
- OOBW

### **MSGDSC (MQMD)-entrada/saída**

Descritor de mensagens

Essa estrutura descreve os atributos da mensagem requerida e os atributos da mensagem recuperada. Consulte [“MQMD \(Descritor de mensagens\) em IBM i”](#) na página 1148 para obter detalhes.

Se BUFLen for menor que o comprimento da mensagem, MSGDSC ainda será inserido pelo gerenciador de filas, se GMATM for especificado no parâmetro **GMO** (consulte o campo GMOPT descrito em [“MQGMO \(opções Get-message\) em IBM i”](#) na página 1113).

Se o aplicativo fornecer um MQMD version-1, a mensagem retornada terá um MQMDE prefixado para os dados da mensagem do aplicativo, mas somente se um ou mais campos no MQMDE tiverem um

valor não padrão. Se todos os campos no MQMDE tiverem valores padrão, MQMDE será omitido. Um nome de formato de FMMDE no campo MDFMT em MQMD indica que um MQMDE está presente

### **GMO (MQGMO)-entrada/saída**

Opções que controlam a ação de MQGET..

Consulte [“MQGMO \(opções Get-message\) em IBM i”](#) na página 1113 para obter detalhes.

### **BUFLEN (número inteiro assinado de 10 dígitos)-entrada**

Comprimento em bytes da área BUFFER .

Zero pode ser especificado para mensagens que não possuem dados ou se a mensagem deve ser removida da fila e os dados descartados (o GMATM deve ser especificado neste caso)

**Nota:** O comprimento da mensagem mais longa que é possível ler na fila é fornecido pelo atributo da fila **MaxMsgLength** ; consulte [“Atributos para filas”](#) na página 1416.

### **BUFFER (cadeia de bits de 1 byte x BUFLEN)-saída**

A área para conter os dados da mensagem

O buffer deve ser alinhado em um limite apropriado para a natureza dos dados na mensagem O alinhamento de 4 bytes deve ser adequado para a maioria das mensagens (incluindo mensagens contendo estruturas de cabeçalho IBM MQ ), mas algumas mensagens podem requerer alinhamento mais rigoroso. Por exemplo, uma mensagem contendo um número inteiro binário de 64 bits pode requerer alinhamento de 8 bytes.

Se BUFLEN for menor que o comprimento da mensagem, a maior parte da mensagem possível será movida para BUFFER ; se o GMATM for especificado no parâmetro **GMO** (consulte o campo GMOPT descrito em [“MQGMO \(opções Get-message\) em IBM i”](#) na página 1113 para obter mais informações).

O conjunto de caracteres e a codificação dos dados em **BUFFER** são fornecidos pelos campos MDCSI e MDENC retornados no parâmetro **MSGDSC** . Se esses valores forem diferentes dos valores requeridos pelo receptor, o receptor deverá converter os dados da mensagem do aplicativo para o conjunto de caracteres e codificação requeridos. A opção GMCONV pode ser usada com uma saída gravada pelo usuário para executar a conversão dos dados da mensagem (consulte [“MQGMO \(opções Get-message\) em IBM i”](#) na página 1113 para obter detalhes dessa opção)

**Nota:** Todos os outros parâmetros na chamada MQGET estão no conjunto de caracteres e na codificação do gerenciador de fila local (fornecido pelo atributo do gerenciador de filas **CodedCharSetId** e ENNAT).

Se a chamada falhar, o conteúdo do buffer pode ainda ter mudado.

### **DATLEN (número inteiro assinado de 10 dígitos)-saída**

O comprimento da mensagem.

Este é o comprimento em bytes dos dados do aplicativo na mensagem. Se esse comprimento de mensagem for maior que BUFLEN, somente BUFLEN bytes serão retornados no parâmetro **BUFFER** (ou seja, a mensagem será truncada). Se o valor for zero, significa que a mensagem não contém dados do aplicativo.

Se BUFLEN for menor que o comprimento da mensagem, DATLEN ainda será inserido pelo gerenciador de filas, se GMATM for especificado no parâmetro **GMO** (consulte o campo GMOPT descrito em [“MQGMO \(opções Get-message\) em IBM i”](#) na página 1113 para obter mais informações). Isso permite que o aplicativo determine o tamanho do buffer necessário para acomodar os dados da mensagem, e depois emita novamente a chamada com um buffer do tamanho apropriado.

No entanto, se a opção GMCONV for especificada e os dados da mensagem convertidos forem muito longos para caber em BUFFER, o valor retornado para DATLEN será:

- O comprimento dos dados não convertidos para formatos definidos pelo gerenciador de filas.



Nesse caso, se a natureza dos dados fizer com que sejam expandidos durante a conversão, o aplicativo deverá alocar um buffer maior que o valor retornado pelo gerenciador de filas para DATLEN.

- O valor retornado pela saída de conversão de dados, para formatos definidos pelo aplicativo.

### **CMPCOD (número inteiro assinado de 10 dígitos)-saída**

Código de conclusão.

É um dos seguintes:

#### **CCOK**

Indica conclusão bem-sucedida.

#### **CCWARN**

Aviso (conclusão parcial).

#### **CCFAIL**

A chamada falhou.

### **REASON (número inteiro assinado de 10 dígitos)-saída**

Código de razão qualificando CMPCOD.

Os seguintes códigos de razão são aqueles que o gerenciador de filas pode retornar para o parâmetro **REASON**. Se o aplicativo especificar a opção GMCONV e uma saída gravada pelo usuário for chamada para converter alguns ou todos os dados da mensagem, será a saída que decide qual valor será retornado para o parâmetro **REASON**. Como resultado, os valores diferentes dos valores documentados posteriormente nesta seção são possíveis.

Se CMPCOD for CCOK:

#### **RCNONE**

(0, X'000') Nenhuma razão a relatar.

Se CMPCOD for CCWARN:

#### **RC2120**

(2120, X'848') Dados convertidos muito grandes para o buffer.

#### **RC2190**

(2190, X'88E') Sequência convertida muito grande para o campo.

#### **RC2150**

(2150, X'866') Sequência DBCS inválida.

#### **RC2110**

(2110, X'83E') Formato da mensagem inválido.

#### **RC2243**

(2243, X'8C3') Segmentos da mensagem possuem CCSIDs diferentes.

#### **RC2244**

(2244, X'8C4') Segmentos da mensagem possuem codificações diferentes.

#### **RC2209**

(2209, X'8A1') Nenhuma mensagem bloqueada.

#### **RC2119**

(2119, X'847') Dados da mensagem não convertidos.

#### **RC2272**

(2272, X'8E0') Dados da mensagem parcialmente convertidos.

#### **RC2145**

(2145, X'861') Parâmetro de buffer de origem inválido.

#### **RC2111**

(2111, X'83F') Identificador do conjunto de caracteres codificados da origem inválido.

#### **RC2113**

(2113, X'841') Codificação decimal compactada na mensagem não reconhecida.

**RC2114**

(2114, X'842') Codificação de ponto flutuante na mensagem não reconhecida.

**RC2112**

(2112, X'840') Codificação de número inteiro da origem não reconhecida.

**RC2143**

(2143, X'85F') Parâmetro de comprimento de origem inválido.

**RC2146**

(2146, X'862') Parâmetro de buffer de destino inválido.

**RC2115**

(2115, X'843') Identificador do conjunto de caracteres codificados do destino inválido.

**RC2117**

(2117, X'845') Codificação decimal compactada especificada pelo receptor não reconhecida.

**RC2118**

(2118, X'846') Codificação de ponto flutuante especificada pelo receptor não reconhecida.

**RC2116**

(2116, X'844') Codificação de número inteiro do destino não reconhecida.

**RC2079**

(2079, X'81F') Mensagem truncada retornada (processamento concluído).

**RC2080**

(2080, X'820') Mensagem truncada retornada (processamento não concluído).

Se CMPCOD for CCFAIL:

**RC2004**

(2004, X'7D4') Parâmetro de buffer inválido.

**RC2005**

(2005, X'7D5') Parâmetro de comprimento de buffer inválido.

**RC2219**

(2219, X'8AB') chamada MQI reinserida antes da chamada anterior ser concluída.

**RC2009**

(2009, X'7D9') Conexão ao gerenciador de filas perdida.

**RC2010**

(2010, X'7DA') Parâmetro de comprimento de dados inválido.

**RC2016**

(2016, X'7E0 ') É inibida para a fila.

**RC2186**

(2186, X'88A') Estrutura de opções de obtenção de mensagem inválida.

**RC2018**

(2018, X'7E2') Manipulação de conexões não válida.

**RC2019**

(2019, X'7E3') Manipulação de objetos não válida.

**RC2241**

(2241, X'8C1') Grupo de mensagens não concluído.

**RC2242**

(2242, X'8C2') Mensagem lógica não concluída.

**RC2259**

(2259, X'8D3') Especificação de pesquisa inconsistente.

**RC2245**

(2245, X'8C5') Especificação de unidade de trabalho inconsistente.

**RC2246**

(2246, X'8C6 ') Mensagem sob o cursor inválida para recuperação.

- RC2247**  
(2247, X'8C7') Opções correspondentes inválidas.
- RC2026**  
(2026, X'7EA') Descritor de mensagem inválido.
- RC2250**  
(2250, X'8CA') Número de sequência da mensagem inválido.
- RC2033**  
(2033, X'7F1') Nenhuma mensagem disponível.
- RC2034**  
(2034, X'7F2') Cursor de navegação não posicionado na mensagem.
- RC2036**  
(2036, X'7F4') Fila não aberta para navegação.
- RC2037**  
(2037, X'7F5') Fila não aberta para entrada.
- RC2041**  
(2041, X'7F9') Definição de objeto alterada desde a abertura.
- RC2101**  
(2101, X'835') Objeto danificado.
- RC2046**  
(2046, X'7FE') Opções não válidas ou não consistentes.
- RC2052**  
(2052, X'804') A fila foi excluída.
- RC2058**  
(2058, X'80A') Nome do gerenciador de filas não válido ou não conhecido.
- RC2059**  
(2059, X'80B') Gerenciador de filas não disponível para conexão.
- RC2161**  
(2161, X'871') Quiesce do gerenciador de filas.
- RC2162**  
(2162, X'872') Gerenciador de filas sendo encerrado.
- RC2102**  
(2102, X'836') Recursos insuficientes do sistema disponíveis.
- RC2071**  
(2071, X'817') Armazenamento insuficiente disponível.
- RC2024**  
(2024, X'7E8') Nenhuma mensagem adicional pode ser manipulada dentro da unidade de trabalho atual.
- RC2072**  
(2072, X'818 ') Suporte de ponto de sincronização não disponível.
- RC2195**  
(2195, X'893') Ocorreu um erro inesperado.
- RC2255**  
(2255, X'8CF') Unidade de trabalho não disponível para o gerenciador de filas a usar.
- RC2090**  
(2090, X'82A') Intervalo de Espera em MQGMO inválido.
- RC2256**  
(2256, X'8D0') Versão errada do MQGMO fornecido.
- RC2257**  
(2257, X'8D1') Versão errada do MQMD fornecido.

## Declaração de RPG

```
C*..1.....2.....3.....4.....5.....6.....7..
C          CALLP      MQGET(HCONN : HOBJ : MSGDSC : GMO :
C          BUFLN : BUFFER : DATLEN :
C          CMPCOD : REASON)
```

A definição de protótipo para a chamada é:

```
D*..1.....2.....3.....4.....5.....6.....7..
DMQGET          PR          EXTPROC('MQGET')
D* Connection handle
D HCONN          10I 0 VALUE
D* Object handle
D HOBJ          10I 0 VALUE
D* Message descriptor
D MSGDSC          364A
D* Options that control the action of MQGET
D GMO          112A
D* Length in bytes of the Buffer area
D BUFLN          10I 0 VALUE
D* Area to contain the message data
D BUFFER          * VALUE
D* Length of the message
D DATLEN          10I 0
D* Completion code
D CMPCOD          10I 0
D* Reason code qualifying CMPCOD
D REASON          10I 0
```

IBM i

## MQINQ (Consultar sobre atributos do objeto) em IBM i

A chamada MQINQ retorna uma matriz de números inteiros e um conjunto de cadeias de caracteres contendo os atributos de um objeto.

Os seguintes tipos de objeto são válidos:

- Fila
- Lista de Nomes
- Definição de processo
- Gerenciador de filas
- [“Sintaxe” na página 1352](#)
- [“Observações de Uso” na página 1352](#)
- [“Parâmetros” na página 1354](#)
- [“Declaração de RPG” na página 1360](#)

### Sintaxe

MQINQ (*HCONN*, *HOBJ*, *SELCNT*, *SELS*, *IACNT*, *INTATR*, *CALEN*, *CHRATR*, *CMPCOD*, *REASON*)

### Observações de Uso

1. Os valores retornados são uma captura instantânea dos atributos selecionados Não há garantia de que os atributos não sejam alterados antes que o aplicativo possa agir nos valores retornados.
2. Ao abrir uma fila modelo, uma fila local dinâmica é criada. Isso é verdadeiro mesmo se você abrir a fila modelo para consultar seus atributos.

Os atributos da fila dinâmica (com determinadas exceções) são os mesmos da fila modelo no momento em que a fila dinâmica é criada. Se você então usar a chamada MQINQ nessa fila, o gerenciador de filas retornará os atributos da fila dinâmica e não aqueles da fila modelo. Consulte a [Tabela 1](#) para obter detalhes de quais atributos da fila modelo são herdados pela fila dinâmica.

3. Se o objeto que está sendo consultado for uma fila de alias, os valores de atributos retornados pela chamada MQINQ serão aqueles da fila de alias e não aqueles da fila de base para a qual o alias é resolvido
4. Se o objeto que está sendo consultado for uma fila de clusters, os atributos que podem ser consultados dependem de como a fila é aberta:
  - Se a fila de clusters for aberta para consulta mais uma ou mais de entrada, navegação ou configuração, deve haver uma instância local da fila de cluster para que a abertura seja bem-sucedida. Neste caso, os atributos que podem ser consultados são aqueles válidos para filas locais
  - Se a fila de clusters for aberta para consulta sozinha ou consulta e saída, apenas os atributos a seguir poderão ser consultados; o atributo **QType** possui o valor QTCLUS neste caso:
    - CAQD
    - CAQN
    - IADBND
    - IADPER
    - IADPRI
    - IAIPUT
    - IAQTYP

Se a fila de clusters for aberta sem nenhuma ligação fixa (ou seja, OOBNDN especificado na chamada MQOPEN ou OOBNDQ especificado quando o atributo **DefBind** tiver o valor BNDNOT), chamadas MQINQ sucessivas para a fila poderão consultar diferentes instâncias da fila de clusters, embora geralmente todas as instâncias tenham os mesmos valores de atributos.

Para obter mais informações sobre filas de clusters, consulte [Configurando um Cluster do Gerenciador de Filas](#)

5. Se um número de atributos tiver que ser consultado e, em seguida, alguns deles devem ser configurados usando a chamada MQSET, pode ser conveniente posicionar no início das matrizes do seletor os atributos que devem ser configurados, para que as mesmas matrizes (com contagens reduzidas) possam ser usadas para MQSET.
6. Se mais de uma das situações de aviso surgir (consulte o parâmetro **CMPCOD**), o código de razão retornado será o *primeiro* na lista a seguir que se aplica:
  - a. RC2068
  - b. RC2022
  - c. RC2008
7. Para obter mais informações sobre atributos de objeto, consulte:
  - [“Atributos para filas” na página 1416](#)
  - [“Atributos para Listas de Nomes” na página 1446](#)
  - [“Atributos para definições de processo em IBM i” na página 1447](#)
  - [“Atributos para o gerenciador de filas no IBM i” na página 1449](#)
8. Uma nova fila local SYSTEM.ADMIN.COMMAND.EVENT é usado para enfileirar mensagens que são geradas sempre que comandos são emitidos. Mensagens são colocadas nesta fila para a maioria dos comandos, dependendo de como o atributo do gerenciador de filas CMDEV é configurado:
  - ENABLED-as mensagens do evento de comando são geradas e colocadas na fila para todos os comandos bem-sucedidos
  - NODISPLAY-as mensagens do evento de comando são geradas e colocadas na fila para todos os comandos bem-sucedidos, exceto o comando DISPLAY (MQSC) e o comando Inquire (PCF.).
  - DISABLED-mensagens do evento de comando não são geradas (este é o valor padrão inicial do gerenciador de filas).

## Parâmetros

A chamada MQINQ possui os parâmetros a seguir:

### **HCONN (número inteiro assinado de 10 dígitos)-entrada**

Manipulação de conexões.

Esta manipulação representa a conexão ao gerenciador de filas. O valor de *HCONN* foi retornado por uma chamada anterior MQCONN ou MQCONNX.

### **HOBJ (número inteiro assinado de 10 dígitos)-entrada**

Manipulação de objetos.

Essa manipulação representa o objeto (de qualquer tipo) com atributos necessários. A manipulação deve ter sido retornada por uma chamada MQOPEN anterior que especificou a opção OOINQ

### **SELCNT (número inteiro assinado de 10 dígitos)-entrada**

Contagem de seletores

Essa é a contagem de seletores fornecidos na matriz *SELS*. É o número de atributo que deve ser retornado. Zero é um valor válido. O número máximo permitido é 256..

### **(Número inteiro assinado de 10 dígitos x SELCNT)-entrada de**

Matriz de seletores de atributo..

Esta é uma matriz de seletores de atributo **SELCNT**; cada seletor identifica um atributo (número inteiro ou caractere) com um valor que é necessário.

Cada seletor deve ser válido para o tipo de objeto que o *HOBJ* representa, caso contrário, a chamada falhará com código de conclusão CCFAIL e código de razão RC2067.

No caso especial de filas:

- Se o seletor não for válido para filas de *qualquer* tipo, a chamada falhará com código de conclusão CCFAIL e código de razão RC2067.
- Se o seletor for aplicável apenas a filas de tipos diferentes do objeto, a chamada será bem-sucedida com o código de conclusão CCWARN e o código de razão RC2068.
- Se a fila que está sendo consultada for uma fila de clusters, os seletores válidos dependem de como a fila foi resolvida; consulte a nota de uso 4 para obter detalhes adicionais.

Os seletores podem ser especificados em qualquer ordem. Os valores de atributos que correspondem aos seletores de atributos de número inteiro (seletores IA\*) são retornados em *INTATR* na mesma ordem em que esses seletores ocorrem em *SELS*. Os valores de atributos que correspondem aos seletores de atributo de caractere (seletores CA\*) são retornados em *CHRATR* na mesma ordem em que esses seletores ocorrem.. Seletores IA\* podem ser intercalados com os seletores CA\*; apenas a ordem relativa dentro de cada tipo é importante.

#### **Nota:**

1. Os seletores de atributo de número inteiro e de caractere são alocados dentro de dois intervalos diferentes; os seletores IA\* residem no intervalo IAFRST até IALAST e os seletores CA\* dentro do intervalo CAFRST até CALAST.

Para cada intervalo, as constantes IALSTU e CALSTU definem o valor mais alto que o gerenciador de filas aceita.

2. Se todos os seletores IA\* ocorrerem primeiro, os mesmos números de elementos poderão ser usados para endereçar os elementos correspondentes nas matrizes *SELS* e *INTATR*.

Os atributos que podem ser consultados são listados nas tabelas a seguir: Para os seletores CA\*, a constante que define o comprimento em bytes da sequência resultante em *CHRATR* é fornecida entre parênteses.

<i>Tabela 746. Seletores de atributo MQINQ para filas</i>		
<b>Seletor</b>	<b>Descrição</b>	<b>Nota</b>
CAALTD	Data da alteração mais recente (LNDATE).	1
CAALTT	Hora da alteração mais recente (LNTIME).	1
CABRQN	Nome do backout-requeue excessivo (LNQN).	5
CABASQ	Nome da fila para a qual o alias é resolvido (LNQN).	
CACFSN	Nome da estrutura do recurso de acoplamento (LNCFSN).	3
CACLN	Nome do cluster (LNCLUN).	1
CACLNL	Lista de nomes do cluster (LNNLN).	1
CACRTD	Data de criação da fila (LNCRTD).	
CACRTT	Horário de criação da fila (LNCRTT).	
CAINIQ	Nome da fila de inicialização (LNQN).	
CAPRON	Nome da definição de processo (LNPRON).	
CAQD	Descrição da fila (LNQD).	
CAQN	Nome da fila (LNQN).	
CARQMN	Nome do gerenciador de fila remota (LNQMN).	
CARQN	Nome da fila remota conforme conhecido no gerenciador de fila remota (LNQN).	
CATRGD	Dados do acionador (LNTRGD).	5
CAXQN	Nome da fila de transmissão (LNQN).	
IABTHR	Limite de restauração.	5
IACDEP	O número de mensagens na fila	
IADBND	Ligação padrão..	1
IADINP	Opção open-for-input padrão.	5
IADPER	Persistência de mensagem padrão..	
IADPRI	Prioridade da mensagem padrão.	5
IADEFT	O tipo de definição de fila.	
IADIST	Suporte à lista de distribuições	2
IAHGB	Se a contagem de backout deve ser endurecida.	5
IAIGET	Se operações get são permitidas.	
IAIPUT	Se operações put são permitidas.	
IAMLEN	Comprimento máximo da mensagem.	
IAMDEP	O número máximo de mensagens permitidas na fila.	
IAMDS	Indica se a prioridade da mensagem é relevante	5
IAOIC	O número de chamadas MQOPEN que têm a fila aberta para entrada	
IAOOC	O número de chamadas MQOPEN que possuem a fila aberta para saída	
IAQDHE	Atributo de controle para eventos de alta profundidade da fila	4, 5

<i>Tabela 746. Seletores de atributo MQINQ para filas (continuação)</i>		
<b>Seletor</b>	<b>Descrição</b>	<b>Nota</b>
IAQDHL	Limite alto para a profundidade da fila	4, 5
IAQDLE	Atributo de controle para eventos de baixa profundidade da fila	4, 5
IAQDLL	Limite baixo para a profundidade da fila	4, 5
IAQDME	Atributo de controle para eventos máximos de profundidade da fila.	4, 5
IAQSI	Limite para intervalo de serviço de fila.	4, 5
IAQSIE	Atributo de controle para eventos de intervalo de serviço de fila	4, 5
IAQTYP	Tipo de fila.	
IAQSGD	Disposição do grupo de compartilhamento de filas	3
IARINT	Intervalo de retenção da fila.	5
IASCOP	Escopo da definição de fila	4, 5
IASHAR	Se a fila pode ser compartilhada para a entrada	
IATRGC	Acionador de controle.	
IATRGD	Profundidade do acionador.	5
IATRGP	Prioridade da mensagem limite para acionadores.	5
IATRGT	Tipo de acionador.	
IAUSAG	Uso.	
CLWLUSEQ	Use as filas remotas	

**Nota:**

1. Suportado nas seguintes plataformas:



-  AIX
-  IBM i
-  Windows
-  z/OS

e para o IBM MQ MQI clients conectado a esses sistemas

2. Suportado nas seguintes plataformas:

-  AIX
-  IBM i
-  Windows

e para clientes IBM MQ conectados a esses sistemas.

3.  Suportado em z/OS
4.  Não suportado em z/OS..
5. Não suportado em VSE/ESA..



<i>Tabela 747. Seletores de atributo MQINQ para listas de nomes</i>		
<b>Seletor</b>	<b>Descrição</b>	<b>Nota</b>
CAALTD	Data da alteração mais recente (LNDATE)	1
CAALTT	Horário da alteração mais recente (LNTIME)	1
CALSTD	Descrição da Lista de Nomes (LNNLD)	1
CALSTN	Nome do objeto da lista de nomes (LNNLN)	1
CANAMS	Nomes na lista de nomes (LNQN x Número de nomes na lista)	1
IANAMC	Número de nomes na lista de nomes	1
IAQSGD	Disposição do Grupo de Compartilhamento de Fila	3

<i>Tabela 748. seletores de atributo MQINQ para definições de processo</i>		
<b>Seletor</b>	<b>Descrição</b>	<b>Nota</b>
CAALTD	Data da alteração mais recente (LNDATE)	1
CAALTT	Horário da alteração mais recente (LNTIME)	1
CAAPPI	Identificador do aplicativo (LNPROA)	5
CAENVD	Dados de ambiente (LNPROE)	5
CAPROD	Descrição da definição de processo (LNPROD)	5
CAPRON	Nome da definição de processo (LNPRON)	5
CAUSRD	Dados do usuário (LNPROU).	5
IAAPPT	Tipo de aplicativo	5
IAQSGD	Disposição do Grupo de Compartilhamento de Fila	3

<i>Tabela 749. seletores de atributos MQINQ para o gerenciador de filas</i>		
<b>Seletor</b>	<b>Descrição</b>	<b>Nota</b>
CAALTD	Data da alteração mais recente (LNDATE)	1
CAALTT	Horário da alteração mais recente (LNTIME)	1
CACADX	Nome de saída de definição de canal automático (LNEXN).	1
CACLWD	Dados transmitidos para a saída de carga de trabalho do cluster (LNEXDA)	1
CACLWX	Nome da saída de carga de trabalho do cluster (LNEXN).	1
CACMDQ	Nome da fila de entrada do comando do sistema (LNQN)	5
CADLQ	Nome da fila de devoluções (LNQN)	5
CADXQN	Nome da fila de transmissão padrão (LNQN)	5
CAQMD	Descrição do gerenciador de filas (LNQMD)	5
CAQMID	Identificador do gerenciador de filas (LNQMID)	1
CAQMN	Nome do gerenciador de filas locais (LNQMN)	5
CAQSGN	Nome do grupo de filas compartilhadas (LNQSGN)	3

<i>Tabela 749. seletores de atributos MQINQ para o gerenciador de filas (continuação)</i>		
<b>Seletor</b>	<b>Descrição</b>	<b>Nota</b>
CARPN	Nome do cluster para o qual o gerenciador de filas fornece serviços de repositório (LNQMN)	1
CARPNL	Nome do objeto da lista de nomes contendo nomes de clusters para os quais o gerenciador de filas fornece serviços de repositório (LNNLN)	1
CMDEV	Atributo de controle que determina se as mensagens geradas quando comandos são emitidos são colocadas em uma fila	8
IAAUTE	Atributo de controle para eventos de autoridade..	4, 5
IACAD	Atributo de controle para definição de canal automático	2
IACADE	Atributo de controle para eventos de definição de canal automática	2
IACLXQ	Tipo de fila de transmissão do cluster padrão	4
IACLWL	Comprimento de carga de trabalho do cluster	1
IACCSI	Identificador do conjunto de caracteres codificados	5
IACMDL	Nível de comando suportado pelo gerenciador de filas	5
IACFGE	Atributo de controle para eventos de configuração.	3
IADIST	Suporte à lista de distribuição	2
IAINHE	Atributo de controle para eventos de inibição	4, 5
IACLE	Atributo de controle para eventos locais	4, 5
IAMHND	Número máximo de identificadores	5
IAMLEN	Comprimento máximo de mensagem	5
IAMPRI	Prioridade máxima	5
IAMUNC	Número máximo de mensagens não confirmadas em uma unidade de trabalho	5
IAPFME	Atributo de controle para eventos de desempenho..	4, 5
IAPLAT	Plataforma na qual o gerenciador de filas reside	5
IARMTE	Atributo de controle para eventos remotos	4, 5
IASSE	Atributo de controle para eventos de parada de início	4, 5
IASYNC	Disponibilidade do ponto de sincronização	5
IATRLFT	Duração de tópicos não administrativos não utilizados	
IATRGI	Intervalo de ativação	5

#### **IACNT (número inteiro assinado de 10 dígitos)-entrada**

Contagem de atributos de número inteiro.

Este é o número de elementos na matriz INTATR .. Zero é um valor válido.

Se for pelo menos o número de seletores IA\* no parâmetro **SELS** , todos os atributos de número inteiro solicitados serão retornados.

#### **INTATR (número inteiro assinado de 10 dígitos x IACNT)-saída**

Matriz de atributos de número inteiro

Essa é uma matriz de valores de atributo de número inteiro *IACNT* .

Os valores de atributo de número inteiro são retornados na mesma ordem dos seletores IA\* no parâmetro **SELS** . Se a matriz contiver mais elementos do que o número de seletores IA\*, os elementos em excesso permanecerão inalterados.

Se HOBJ representar uma fila, mas um seletor de atributo não for aplicável a esse tipo de fila, o valor específico IAVNA será retornado para o elemento correspondente na matriz INTATR .

#### **CALEN (número inteiro assinado de 10 dígitos)-entrada**

O comprimento do buffer de atributos de caracteres

Esse é o comprimento em bytes do parâmetro **CHRATR** .

Deve ser pelo menos a soma dos comprimentos dos atributos de caractere solicitados (consulte SELS). Zero é um valor válido.

#### **CHRATR (1 byte cadeia de caracteres x CALEN)-saída**

Atributos de caractere..

Este é o buffer no qual os atributos de caractere são retornados, concatenados juntos O comprimento do buffer é fornecido pelo parâmetro **CALEN** .

Os atributos de caractere são retornados na mesma ordem dos seletores CA\* no parâmetro **SELS** . O comprimento de cada sequência de atributos é fixo para cada atributo (consulte SELS) e o valor nele é preenchido à direita com espaços em branco se necessário. Se o buffer for maior do que o necessário para conter todos os atributos de caracteres solicitados (incluindo preenchimento), os bytes além do último valor de atributo retornado permanecerão inalterados

Se HOBJ representar uma fila, mas um seletor de atributos não for aplicável a esse tipo de fila, uma sequência de caracteres que consiste inteiramente em asteriscos (\*) será retornado como o valor desse atributo em CHRATR.

#### **CMPCOD (número inteiro assinado de 10 dígitos)-saída**

Código de conclusão.

É um dos seguintes:

##### **CCOK**

Indica conclusão bem-sucedida.

##### **CCWARN**

Aviso (conclusão parcial).

##### **CCFAIL**

A chamada falhou.

#### **REASON (número inteiro assinado de 10 dígitos)-saída**

Código de razão qualificando CMPCOD.

Se CMPCOD for CCOK:

##### **RCNONE**

(0, X'000') Nenhuma razão a relatar.

Se *CMPCOD* for CCWARN:

##### **RC2008**

(2008, X'7D8') Não há espaço suficiente permitido para atributos de caracteres.

##### **RC2022**

(2022, X'7E6') Não há espaço suficiente permitido para atributos de número inteiro.

##### **RC2068**

(2068, X'814 ') Selector não aplicável ao tipo de fila.

Se *CMPCOD* for CCFAIL:

##### **RC2219**

(2219, X'8AB') chamada MQI reinserida antes da chamada anterior ser concluída.

- RC2006**  
(2006, X'7D6') Comprimento de atributos de caractere não válido.
- RC2007**  
(2007, X'7D7') Sequência de atributos de caractere não válida.
- RC2009**  
(2009, X'7D9') Conexão ao gerenciador de filas perdida.
- RC2018**  
(2018, X'7E2') Manipulação de conexões não válida.
- RC2019**  
(2019, X'7E3') Manipulação de objetos não válida.
- RC2021**  
(2021, X'7E5') Contagem de atributos de número inteiro inválida.
- RC2023**  
(2023, X'7E7') A matriz de atributos de número inteiro não é válida.
- RC2038**  
(2038, X'7F6') Fila não aberta para consulta.
- RC2041**  
(2041, X'7F9') Definição de objeto alterada desde a abertura.
- RC2101**  
(2101, X'835') Objeto danificado.
- RC2052**  
(2052, X'804') A fila foi excluída.
- RC2058**  
(2058, X'80A') Nome do gerenciador de filas não válido ou não conhecido.
- RC2059**  
(2059, X'80B') Gerenciador de filas não disponível para conexão.
- RC2162**  
(2162, X'872') Gerenciador de filas sendo encerrado.
- RC2102**  
(2102, X'836') Recursos insuficientes do sistema disponíveis.
- RC2065**  
(2065, X'811 ') Contagem de seletores inválida.
- RC2067**  
(2067, X'813 ') Seletor de atributo inválido.
- RC2066**  
(2066, X'812 ') Contagem de seletores muito grande.
- RC2071**  
(2071, X'817') Armazenamento insuficiente disponível.
- RC2195**  
(2195, X'893') Ocorreu um erro inesperado.

## Declaração de RPG

```

C*..1.....2.....3.....4.....5.....6.....7..
C          CALLP      MQINQ(HCONN : HOBJ : SELCNT :
C          SELS(1) : IACNT : INTATR(1) :
C          CALEN : CHRATR : CMPCOD :
C          REASON)

```

A definição de protótipo para a chamada é:

```

D*..1.....2.....3.....4.....5.....6.....7..
DMQINQ          PR          EXTPROC('MQINQ')
D* Connection handle
D HCONN          10I 0 VALUE
D* Object handle
D HOBJ          10I 0 VALUE
D* Count of selectors
D SELCNT        10I 0 VALUE
D* Array of attribute selectors
D SELS          10I 0
D* Count of integer attributes
D IACNT         10I 0 VALUE
D* Array of integer attributes
D INTATR        10I 0
D* Length of character attributes buffer
D CALEN        10I 0 VALUE
D* Character attributes
D CHRATR          *   VALUE
D* Completion code
D CMPCOD        10I 0
D* Reason code qualifying CMPCOD
D REASON        10I 0

```

## IBM i MQINQMP (Inquire message property) em IBM i

A chamada MQINQMP retorna o valor de uma propriedade de uma mensagem.

- [“Sintaxe” na página 1361](#)
- [“Parâmetros” na página 1361](#)
- [“Declaração de RPG” na página 1365](#)

### Sintaxe

MQINQMP (*Hconn*, *Hmsg*, *InqPropOpts*, *Name*, *PropDesc*, *Type*, *ValueLength*, *Value*, *DataLength*, *CompCode*, *Reason*)

### Parâmetros

A chamada MQINQMP possui os parâmetros a seguir:

#### HCONN (número inteiro assinado de 10 dígitos)-entrada

Esta manipulação representa a conexão ao gerenciador de filas. O valor de *Hconn* deve corresponder à manipulação de conexões que foi usada para criar a manipulação de mensagens especificada no parâmetro **Hmsg** ..

Se o identificador de mensagens tiver sido criado usando HCUNAS, uma conexão válida deverá ser estabelecida no encadeamento consultando uma propriedade do identificador de mensagens, caso contrário, a chamada falhará com RC2009

#### HMSG (número inteiro assinado de 20 dígitos)-entrada

Esta é a manipulação de mensagem a ser consultada. O valor foi retornado por uma chamada anterior de **MQCRTMH**

#### INQOPT (MQIMPO)-entrada

Consulte o tipo de dados [MQIMPO](#) para obter detalhes..

#### PRNAME (MQCHARV)-entrada

Isso descreve o nome da propriedade para consultar.

Se não for possível localizar nenhuma propriedade com esse nome, a chamada falhará com o motivo RC2471

É possível usar o caractere de sinal de porcentagem (%) no final do nome da propriedade. O curinga corresponde a zero ou mais caracteres, incluindo o caractere ponto (.). Isso permite que um aplicativo

inquira o valor de muitas propriedades. Chame MQINQMP com a opção IPINQF para obter a primeira propriedade correspondente e novamente com a opção IPINQN para obter a próxima propriedade correspondente. Quando mais nenhuma propriedade correspondente estiver disponível, a chamada falhará com RC2471 Se o campo *ReturnedName* da estrutura InqPropOpts for inicializado com um endereço ou deslocamento para o nome retornado da propriedade, isso será concluído no retorno de MQINQMP com o nome da propriedade correspondente. Se o campo *VSBufSize* do *ReturnedName* na estrutura de Opções InqPropfor menor que o comprimento do nome da propriedade retornado, o código de conclusão será configurado como CCFAIL com a razão RC2465.

Propriedades que possuem sinônimos conhecidos são retornadas da seguinte forma:

1. Propriedades com o prefixo "mqps." São retornadas com o nome da propriedade IBM MQ Por exemplo, "MQTopicString" é o nome retornado em vez de "mqps.Top".
2. Propriedades com o prefixo "jms." ou "McD". são retornados como o nome do campo de cabeçalho JMS Por exemplo, "JMSExpiration" é o nome retornado em vez de "jms.Exp".
3. Propriedades com o prefixo "usr." são retornados sem esse prefixo Por exemplo, "Color" é retornado em vez de "usr.Color".

As propriedades com sinônimos são retornadas apenas uma vez

Na linguagem de programação RPG, as seguintes variáveis de macro são definidas para consultar todas as propriedades e todas as propriedades que começam com "usr.":

#### **INQALL**

Consultar em todas as propriedades da mensagem

#### **INQUSR**

Consulte todas as propriedades da mensagem que iniciam "usr.". O nome retornado é retornado sem o "usr." .

Se IPINQN for especificado, mas o Nome foi alterado desde a chamada anterior ou esta é a primeira chamada, então IPINQF será implícito.

Consulte [Nomes de Propriedade e Restrições de Nome de Propriedade](#) para obter informações adicionais sobre o uso de nomes de propriedade

#### **PRPDSC (MQPD)-saída**

Essa estrutura é utilizada para definir os atributos de uma propriedade, incluindo o que acontece se a propriedade não for suportada, a qual contexto de mensagem a propriedade pertence e para quais mensagens a propriedade deve ser copiada. Consulte [MQPD](#) para obter detalhes dessa estrutura..

#### **TYPE (número inteiro assinado de 10 dígitos)-entrada/saída**

No retorno da chamada MQINQMP, esse parâmetro é configurado para o tipo de dados de *Valor* O tipo de dados pode ser qualquer um dos seguintes:

##### **TYPBOL**

Um booleano.

##### **TYPBST**

Uma cadeia de bytes

##### **TYPI8**

Um número inteiro assinado de 8 bits.

##### **TYPI16**

Um número inteiro assinado de 16 bits.

##### **TYPI32**

Um número inteiro assinado de 32 bits.

##### **TYPI64**

Um número inteiro assinado de 64 bits.

##### **TYPF32**

Um número de vírgula flutuante de 32 bits..

**TYPF64**

Um número de vírgula flutuante de 64 bits..

**TYPSTR**

Uma sequência de Caracteres

**TYPNUL**

A propriedade existe mas possui um valor nulo.

Se o tipo de dados do valor da propriedade não for reconhecido, então TYPSTR será retornado e uma representação em cadeia do valor será colocada na área *Valor* Uma representação em sequência do tipo de dados pode ser localizada no campo *IPTYP* do parâmetro *IPOPT* Um código de conclusão de aviso é retornado com o motivo RC2467

Além disso, se a opção IPCTYP for especificada, a conversão do valor da propriedade será solicitada.. Use *Tipo* como uma entrada para especificar o tipo de dados como o qual você deseja que a propriedade seja retornada. Consulte a descrição da opção IPCTYP do “MQIMPO (Consultar opções de propriedade de mensagem) no IBM i” na página 1141 para obter detalhes da conversão de tipo de dados

Se você não solicitar a conversão de tipo, poderá usar o seguinte valor na entrada:

**TYPAST**

O valor da propriedade é retornado sem converter seu tipo de dados.

**VALLEN (número inteiro assinado de 10 dígitos)-entrada**

O comprimento em bytes da área *Valor*.

Especifique zero para propriedades para as quais não é necessário o valor retornado. Elas podem ser propriedades projetadas por um aplicativo para ter um valor nulo ou uma sequência vazia. Especifique também zero se a opção IPQLEN tiver sido especificada; neste caso, nenhum valor será retornado.

**VALUE (bit de 1 byte stringxVALLEN)-saída.**

Essa é a área para conter o valor da propriedade consultada O buffer deve ser alinhado em um limite apropriado para o valor retornado. Não fazer isso pode resultar em um erro quando o valor for acessado posteriormente.

Se *VALLEN* for menor que o comprimento do valor da propriedade, tanto do valor da propriedade quanto possível será movido para *VALUE* e a chamada falhará com o código de conclusão CCFAIL e a razão RC2469.

O conjunto de caracteres dos dados em *VALUE* é fornecido pelo campo IPRETCSI no parâmetro INQOPT. A codificação dos dados em *VALUE* é fornecida pelo campo IPRETENC no parâmetro INQOPT.

Se o parâmetro *VALLEN* for zero, *VALUE* não será referido.

**DATLEN (número inteiro assinado de 10 dígitos)-saída**

Esse é o comprimento em bytes do valor da propriedade real, conforme retornado na área *Valor* .

Se *DataLength* for menor que o comprimento do valor da propriedade, *DataLength* ainda será inserido no retorno da chamada MQINQMP. Isso permite que o aplicativo determine o tamanho do buffer necessário para acomodar o valor da propriedade e, em seguida, emita novamente a chamada com um buffer do tamanho apropriado..

Os seguintes valores também podem ser retornados.

Se o parâmetro *Type* for configurado como TYPSTR ou TYPBST:

**VLEMP**

A propriedade existe mas não contém caracteres ou bytes.

**CMPCOD (número inteiro assinado de 10 dígitos)-saída**

O código de conclusão; é um dos seguintes:

**CCOK**

Indica conclusão bem-sucedida.

**CCWARN**

Aviso (conclusão parcial).

**CCFAIL**

A chamada falhou.

**REASON (número inteiro assinado de 10 dígitos)-saída**

O código de razão que qualifica *CompCode*.

Se *CMPCOD* for CCOK:

**RCNONE**

(0, X'000') Nenhuma razão a relatar.

Se *CompCode* for CCWARN:

**RC2492**

(2492, X'09BC') Nome da propriedade retornada não convertido.

**RC2466**

(2466, X'09A2') Valor da propriedade não convertido.

**RC2467**

(2467, X'09A3') O tipo de dados da propriedade não é suportado

**RC2421**

(2421, X'0975 ') Uma pasta MQRFH2 contendo propriedades não pôde ser analisada.

Se *CMPCOD* for CCFAIL:

**RC2204**

(2204, X'089C') Adaptador não disponível.

**RC2130**

(2130, X'0852 ') Impossível carregar o módulo de serviço do adaptador.

**RC2157**

(2157, X'086D') Os ASIDs primário e inicial diferem.

**RC2004**

(2004, X'07D4') Parâmetro de valor inválido.

**RC2005**

(2005, X'07D5') Parâmetro de comprimento do valor inválido.

**RC2219**

(2219, X'08AB') chamada MQI inserida antes da chamada anterior ser concluída.

**RC2009**

(2009, X'07D9') Conexão com o gerenciador de filas perdida.

**RC2010**

(2010, X'07DA') Parâmetro de comprimento de dados inválido.

**RC2464**

(2464, X'09A0') A estrutura de opções da propriedade de mensagem não é válida.

**RC2460**

(2460, X'099C') Identificador de mensagem inválido.

**RC2499**

(2499, X'09C3') Identificador de mensagem já em uso.

**RC2064**

(2046, X'07F8') Opções não válidas ou não consistentes.

**RC2482**

(2482, X'09B2') Estrutura do descritor de propriedade inválida.



**RC2470**

(2470, X'09A6') Conversão do tipo de dados real para solicitado não suportada.

**RC2442**

(2442, X'098A') Nome da propriedade inválido

**RC2465**

(2465, X'09A1') Nome da propriedade muito grande para o buffer de nome retornado.

**RC2471**

(2471, X'09A7) Propriedade não disponível.

**RC2469**

(2469, X'09A5') Valor da propriedade muito grande para a área Valor.

**RC2472**

(2472, X'09A8') Erro de formato numérico encontrado nos dados de valor...

**RC2473**

(2473, X'09A9') Tipo de propriedade solicitado inválido..

**RC2111**

(2111, X'083F') Identificador do conjunto de caracteres codificados do nome da propriedade inválido.

**RC2071**

(2071, X'0871 ') Armazenamento insuficiente disponível.

**RC2195**

(2195, X'0893 ') Ocorreu um erro inesperado.

Para obter informações detalhadas sobre esses códigos, consulte:

- [IBM MQ for z/OS mensagens, conclusão, e códigos de razão](#) for IBM MQ for z/OS
- [Mensagens e códigos de razão](#) para todas as outras plataformas IBM MQ

## Declaração de RPG

```

C*..1.....2.....3.....4.....5.....6.....7..
C          CALLP      MQINQMP(HCONN : HMSG : INQOPT :
                          PRNAME : PRPDSC : TYPE :
                          VALLEN : VALUE : DATLEN :
                          CMPCOD : REASON)

```

A definição de protótipo para a chamada é:

```

DMQINQMP          PR          EXTPROC('MQINQMP')
D* Connection handle
D HCONN          10I 0 VALUE
D* Message handle
D HMSG          20I 0 VALUE
D* Options that control the action of MQINQMP
D INQOPT          72A
D* Property name
D PRNAME          32A
D* Property descriptor
D PRPDSC          24A
D* Property data type
D TYPE          10I 0
D* Length in bytes of the Value area
D VALLEN          10I 0 VALUE
D* Property value
D VALUE          * VALUE
D* Length of the property value
D DATLEN          10I 0
D* Completion code
D CMPCOD          10I 0
D* Reason code qualifying CompCode
D REASON          10I 0

```

**IBM i**

O MQMHBUF converte um identificador de mensagem em um buffer e é o inverso da chamada de MQBUFMH

- [“Sintaxe” na página 1366](#)
- [“Observações de Uso” na página 1366](#)
- [“Parâmetros” na página 1366](#)
- [“Declaração de RPG” na página 1368](#)

**Sintaxe**

MQMHBUF (*Hconn, Hmsg, MsgHBufOpts, Name, MsgDesc, BufferLength, Buffer, DataLength, CompCode, Reason*)

**Observações de Uso**

O MQMHBUF converte um identificador de mensagem em um buffer

É possível usá-lo com uma saída de API MQGET para acessar determinadas propriedades, usando as APIs de propriedade de mensagens e, em seguida, passado essas propriedades em um buffer de volta para um aplicativo projetado para usar cabeçalhos MQRFH2 em vez de manipulações de mensagens.

Essa chamada é o inverso da chamada MQBUFMH, que pode ser usada para analisar as propriedades de mensagem de um buffer em uma manipulação de mensagens

**Parâmetros**

A chamada MQMHBUF possui os parâmetros a seguir:

**HCONN (número inteiro assinado de 10 dígitos)-entrada**

Esta manipulação representa a conexão ao gerenciador de filas.

O valor de *HCONN* deve corresponder à manipulação de conexões que foi usada para criar a manipulação de mensagens especificada no parâmetro **HMSG** ..

Se o identificador de mensagem foi criado usando HCUNAS, uma conexão válida deverá ser estabelecida no encadeamento excluindo o identificador de mensagem. Se uma conexão válida não for estabelecida a chamada falhará com RC2009.

**HMSG (número inteiro assinado de 20 dígitos)-entrada**

Esse identificador é o identificador de mensagem para o qual um buffer é necessário

O valor foi retornado por uma chamada MQCRTMH anterior.

**MHBOPT (MQMHBO)-entrada**

A estrutura MQMHBO permite que os aplicativos especifiquem opções que controlam como os buffers são produzidos a partir de manipulações de mensagens

Consulte [“MQBMHO \(Buffer para opções de manipulação de mensagens\) no IBM i” na página 1052](#) para obter detalhes.

**PRNAME (MQCHARV)-entrada**

O nome da propriedade ou propriedades a serem colocadas no buffer.

Se nenhuma propriedade correspondente ao nome puder ser localizada, a chamada falhará com RC2471..

**Curingas**

É possível usar um curinga para colocar mais de uma propriedade no buffer. Para isso, use o sinal de porcentagem (%) no final do nome da propriedade. Este curinga corresponde a zero ou mais caracteres, incluindo o caractere de ponto (.).

Consulte [Nomes de Propriedade e Restrições de Nome de Propriedade](#) para obter informações adicionais sobre o uso de nomes de propriedade.

### **MSGDSC (MQMD)-entrada/saída**

A estrutura *MSGDSC* descreve o conteúdo da Área de Buffer.

Na saída, os campos *Encoding*, *CodedCharSetId* e *Format* são configurados para descrever corretamente a codificação, o identificador do conjunto de caracteres e o formato dos dados na área de buffer, conforme gravados pela chamada.

Os dados nessa estrutura estão no conjunto de caracteres e na codificação do aplicativo.

### **BUFLEN (número inteiro assinado de 10 dígitos)-entrada**

*BUFLEN* é o comprimento da área Buffer, em bytes.

### **BUFFER (cadeia de bits de 1 byte x BUFLEN)-entrada/saída**

*BUFFER* define a área contendo o buffer de mensagem. Para a maioria dos dados, você deve alinhar o buffer em um limite de 4 bytes.

Se o *BUFFER* contiver dados de caractere ou numéricos, configure os campos *CodedCharSetId* e *Encoding* no parâmetro **MSGDSC** para os valores apropriados para os dados; isso permite que os dados sejam convertidos, se necessário.

Se as propriedades forem localizadas no buffer de mensagem, elas serão removidas opcionalmente; posteriormente, elas se tornarão disponíveis a partir do identificador de mensagem no retorno da chamada.

Na linguagem de programação C, o parâmetro é declarado como um ponteiro para vazio, o que significa que o endereço de qualquer tipo de dados pode ser especificado como o parâmetro.

Se o parâmetro **BUFLEN** for zero, *BUFFER* não será referido. Nesse caso, o endereço de parâmetro transmitido por programas gravados em C ou assembler System/390 pode ser nulo.

### **DATLEN (número inteiro assinado de 10 dígitos)-saída**

*DATLEN* é o comprimento, em bytes, das propriedades retornadas no buffer. Se o valor for zero, nenhuma propriedade correspondeu ao valor fornecido em *PRNAME* e a chamada falha com o código de razão RC2471.

Se *BUFLEN* for menor que o comprimento necessário para armazenar as propriedades no buffer, a chamada *MQMHBUF* falhará com RC2469, mas um valor ainda será inserido em *DATLEN*. Isso permite que o aplicativo determine o tamanho do buffer necessário para acomodar as propriedades e, em seguida, emita novamente a chamada com o *BUFLEN* necessário.

### **CMPCOD (número inteiro assinado de 10 dígitos)-saída**

O código de conclusão; é um dos seguintes:

#### **CCOK**

Indica conclusão bem-sucedida.

#### **CCFAIL**

A chamada falhou.

### **REASON (número inteiro assinado de 10 dígitos)-saída**

O código de razão qualificando *CMPCOD*.

Se *CMPCOD* for CCOK:

#### **RCNONE**

(0, X'000') Nenhuma razão a relatar.

Se *CMPCOD* for *CCFAIL*:

**RC2204**

(2204, X'089C') Adaptador não disponível.

**RC2130**

(2130, X'852') Não foi possível carregar o módulo de serviço do adaptador.

**RC2157**

(2157, X'86D') ASIDs de início e primárias são diferentes.

**RC2501**

(2501, X'095C') O identificador de mensagens para a estrutura de opções de buffer não é válido..

**RC2004**

(2004, X'07D4') Parâmetro de buffer inválido.

**RC2005**

(2005, X'07D5') Parâmetro de comprimento do buffer inválido.

**RC2219**

(2219, X'08AB') chamada MQI inserida antes da chamada anterior ser concluída.

**RC2009**

(2009, X'07D9') Conexão com o gerenciador de filas perdida.

**RC2010**

(2010, X'07DA') Parâmetro de comprimento de dados inválido.

**RC2460**

(2460, X'099C') Identificador de mensagem inválido.

**RC2026**

(2026, X'07EA') Descritor de mensagens inválido.

**RC2499**

(2499, X'09C3') Identificador de mensagem já em uso.

**RC2046**

(2046, X'07FE') Opções não válidas ou não consistentes.

**RC2442**

(2442, X'098A') O nome da propriedade não é válido.

**RC2471**

(2471, X'09A7') Propriedade não disponível.

**RC2469**

(2469, X'09A5') BufferLength valor é muito pequeno para conter propriedades especificadas.

**RC2195**

(2195, X'893') Ocorreu um erro inesperado.

## Declaração de RPG

```
C*..1.....2.....3.....4.....5.....6.....7..
C          CALLP      MQMHBUF(HCONN : HMSG : MHBOPT :
                          PRNAME : MSGDSC : BUFLen :
                          BUFFER : DATLEN :
                          CMPCOD : REASON)
```

A definição de protótipo para a chamada é:

```
DMQMHBUF          PR          EXTPROC('MQMHBUF')
D* Connection handle
D HCONN           10I 0 VALUE
D* Message handle
D HMSG           20I 0 VALUE
D* Options that control the action of MQMHBUF
D MHBOPT         12A
D* Property name
```

D PRNAME	32A
D* Message descriptor	
D MSGDSC	364A
D* Length in bytes of the Buffer area	
D BUFLN	10I 0 VALUE
D* Area to contain the properties	
D BUFFER	* VALUE
D* Length of the properties	
D DATLEN	10I 0
D* Completion code	
D CMPCOD	10I 0
D* Reason code qualifying CompCode	
D REASON	10I 0

## IBM i MQOPEN (objeto aberto) no IBM i

A chamada MQOPEN estabelece acesso a um objeto.

Os seguintes tipos de objeto são válidos:

- Fila (incluindo listas de distribuição)
- Lista de Nomes
- Definição de processo
- Gerenciador de filas
- Tópico

### Index

- [“Sintaxe” na página 1369](#)
- [“Observações de Uso” na página 1369](#)
- [“Parâmetros” na página 1374](#)
- [“Declaração de RPG” na página 1380](#)

### Sintaxe

MQOPEN (*HCONN, OBJDSC, OPTS, HOBJ, CMPCOD, REASON*)

### Observações de Uso

1. O objeto aberto é um dos seguintes:

- Uma fila, para:
  - Obter ou procurar mensagens (usando a chamada MQGET)
  - Mensagens put (usando a chamada MQPUT)
  - Consultar os atributos da fila (usando a chamada MQINQ)
  - Configure os atributos da fila (usando a chamada MQSET)..

Se a fila denominada for uma fila modelo, uma fila local dinâmica será criada.

Uma lista de distribuição é um tipo especial de objeto de fila que contém uma lista de filas.. Pode ser aberto para colocar mensagens, mas não para obter ou procurar mensagens ou para consultar ou configurar atributos. Consulte a nota de uso 8 para obter detalhes adicionais

Uma fila que tem QSGDISP (GROUP) é um tipo especial de definição de fila que não pode ser usado com as chamadas MQOPEN ou MQPUT1

- Uma lista de nomes, a fim de:
  - Consulte os nomes das filas na lista (usando a chamada MQINQ).
- Uma definição de processo, a fim de:
  - Consulte sobre os atributos de processo (usando a chamada MQINQ)

- O gerenciador de filas, para:
  - Consulte os atributos do gerenciador de fila local (usando a chamada MQINQ).
- 2. É válido para um aplicativo abrir o mesmo objeto mais de uma vez. Uma manipulação de objetos diferente é retornada para cada abertura. Cada identificador que é retornado pode ser usado para as funções para as quais a abertura correspondente foi executada.
- 3. Se o objeto que está sendo aberto for uma fila, mas não uma fila de clusters, toda a resolução de nome dentro do gerenciador de fila local ocorrerá no momento da chamada MQOPEN. Isso pode incluir um ou mais dos seguintes para uma chamada MQOPEN específica:
  - Resolução de alias para o nome de uma fila base
  - Resolução do nome de uma definição local de uma fila remota para o nome do gerenciador de fila remota e o nome pelo qual a fila é conhecida no gerenciador de filas remotas
  - Resolução do nome do gerenciador de filas remotas para o nome de uma fila de transmissão local

No entanto, esteja ciente de que as chamadas MQINQ ou MQSET subsequentes para o identificador estão relacionadas exclusivamente ao nome que foi aberto e não ao objeto resultante após a resolução do nome ter ocorrido. Por exemplo, se o objeto aberto for um alias, os atributos retornados pela chamada MQINQ serão os atributos do alias, não os atributos da fila de base para os quais o alias é resolvido. A verificação de resolução de nome ainda é realizada, no entanto, independentemente do que é especificado para o parâmetro **OPTS** no MQOPEN correspondente.

Se o objeto que está sendo aberto for uma fila de clusters, a resolução do nome poderá ocorrer no momento da chamada MQOPEN ou ser adiada até mais tarde... O ponto no qual a resolução ocorre é controlado pelas opções OOBND\* especificadas na chamada MQOPEN:

- OOBND0
- OOBNDN
- OOBNDQ

Consulte [Resolução de nome](#) para obter mais informações sobre a resolução de nome para filas de cluster...

4. Os atributos de um objeto podem ser alterados enquanto um aplicativo tiver o objeto aberto. Em muitos casos, o aplicativo não percebe isso, mas para determinados atributos, o gerenciador de filas marca o identificador como não mais válido. São elas:
  - Qualquer atributo que afete a resolução do nome do objeto. Isso se aplica independentemente das opções de abertura usadas e inclui o seguinte:
    - Uma mudança para o atributo **BaseQName** de uma fila de alias que está aberta
    - Uma mudança nos atributos da fila **RemoteQName** ou **RemoteQMgrName**, para qualquer identificador que esteja aberto para essa fila, ou para uma fila que seja resolvida por meio dessa definição como um alias do gerenciador de filas
    - Qualquer mudança que faça com que uma manipulação atualmente aberta para uma fila remota seja resolvida para uma fila de *transmissão* diferente ou falhe ao resolver para uma. Por exemplo, isso pode incluir:
      - Uma mudança no atributo **XmitQName** da definição local de uma fila remota, independentemente de a definição estar sendo usada para uma fila ou para um alias do gerenciador de filas

Há uma exceção a isso, a criação de uma nova fila de transmissão. Um identificador que teria sido resolvido para esta fila se estivesse presente quando o identificador foi aberto, mas, em vez disso, resolvido para a fila de transmissão padrão, não é inválido.
  - Uma mudança no atributo do gerenciador de filas do **DefXmitQName**. Nesse caso, todos os identificadores abertos que foram resolvidos para a fila nomeada anteriormente (que foram resolvidos para ela apenas porque era a fila de transmissão padrão) são marcados como inválidos. As manipulações que foram resolvidas para essa fila por outras razões não são afetadas.

- O atributo da fila **Shareability** , se houver duas ou mais manipulações que estão fornecendo atualmente acesso OOINPS para essa fila ou para uma fila que é resolvida para essa fila Em caso afirmativo, *todos* os identificadores que estão abertos para esta fila ou para uma fila que é resolvida para esta fila são marcados como inválidos, independentemente das opções de abertura
- O atributo da fila **Usage** , para todas as manipulações que estão abertas para esta fila ou para uma fila que é resolvida para esta fila, independentemente das opções de abertura

Quando um identificador é marcado como inválido, todas as chamadas subsequentes (diferentes de MQCLOSE) que usam esse manipulador falham com o código de razão RC2041; o aplicativo deve emitir uma chamada MQCLOSE (usando o identificador original) e, em seguida, reabrir a fila. Quaisquer atualizações não confirmadas com relação à manipulação antiga de chamadas bem-sucedidas anteriores ainda podem ser confirmadas ou restauradas, conforme necessário pela lógica de aplicativo

Se a mudança de um atributo causar isso, uma versão "force" especial do comando deverá ser usada.

5. O gerenciador de filas executa verificações de segurança quando uma chamada MQOPEN é emitida, para verificar se o identificador de usuário sob o qual o aplicativo está em execução possui o nível apropriado de autoridade antes que o acesso seja permitido A verificação de autoridade é feita no nome do objeto que está sendo aberto e não no nome, ou nomes, resultante após um nome ter sido resolvido.

Se o objeto que está sendo aberto for uma fila modelo, o gerenciador de filas executará uma verificação de segurança completa com relação ao nome da fila modelo e ao nome da fila dinâmica criada. Se a fila dinâmica resultante for aberta explicitamente, uma verificação de segurança de recurso adicional será executada com relação ao nome da fila dinâmica.

6. Uma fila remota pode ser especificada de uma de duas maneiras no parâmetro **OBJDSC** desta chamada (consulte os campos *ODON* e *ODMN* descritos em [“MQOD \(descriptor de objeto\) em IBM i”](#) na [página 1200](#)):

- Especificando para *ODON* o nome de uma definição local da fila remota. Neste caso, *ODMN* refere-se ao gerenciador de filas locais e pode ser especificado como espaços em branco

A validação de segurança executada pelo gerenciador de fila local verifica se o usuário está autorizado a abrir a definição local da fila remota

- Especificando para *ODON* o nome da fila remota, conforme conhecido para o gerenciador de filas remotas Nesse caso, *ODMN* é o nome do gerenciador de fila remoto

A validação de segurança executada pelo gerenciador de filas locais verifica se o usuário está autorizado a enviar mensagens para a fila de transmissão resultante do processo de resolução de nome

Em ambos os casos:

- Nenhuma mensagem é enviada pelo gerenciador da fila local para o gerenciador de filas remotas para verificar se o usuário está autorizado a colocar mensagens na fila
  - Quando uma mensagem chega ao gerenciador de filas remotas, o gerenciador de filas remotas pode rejeitá-lo porque o usuário que está originando a mensagem não está autorizado.
7. Uma chamada MQOPEN com a opção OOBROW estabelece um cursor de navegação, para uso com chamadas MQGET que especificam a manipulação de objetos e uma das opções de navegação Isso permite que a fila seja digitalizada sem alterar seu conteúdo. Uma mensagem que foi localizada pela navegação pode ser removida posteriormente da fila usando a opção GMMUC.

Vários cursores de navegação podem estar ativos para um único aplicativo emitindo várias solicitações MQOPEN para a mesma fila..

8. As seguintes notas aplicam-se ao uso de listas de distribuição.

- Os campos na estrutura MQOD devem ser configurados conforme a seguir ao abrir uma lista de distribuição:
  - *ODVER* deve ser ODVER2 ou superior.

- *ODOT* deve ser OTQ.
- *ODON* deve estar em branco ou a sequência nula.
- *ODMN* deve estar em branco ou a sequência nula.
- *ODREC* Deve ser maior que zero.
- Um de *ODORO* e *ODORP* deve ser zero e o outro diferente de zero.
- No máximo um de *ODRRO* e *ODRRP* pode ser diferente de zero.
- Deve haver registros de objeto *ODREC* , endereçados por *ODORO* ou *ODORP* Os registros do objeto devem ser configurados para os nomes das filas de destino a serem abertas
- Se um de *ODRRO* e *ODRRP* for diferente de zero, deverá haver *ODREC* registros de resposta presentes... Eles serão configurados pelo gerenciador de filas se a chamada for concluída com o código de razão RC2136..

Um MQOD version-2 também pode ser usado para abrir uma única fila que não esteja em uma lista de distribuição, assegurando que *ODREC* seja zero.

- Apenas as opções de abertura a seguir são válidas no parâmetro **OPTS** :
  - OOOOUT
  - OOPAS\*
  - OOSSET\*
  - OOALTU
  - OOFIQ
- As filas de destino na lista de distribuição podem ser filas locais, de alias ou remotas, mas não podem ser filas modelo. Se uma fila modelo for especificada, essa fila falhará ao abrir, com o código de razão RC2057. No entanto, isso não evita que outras filas na lista sejam abertas com êxito
- Os parâmetros de código de conclusão e de código de razão são configurados conforme a seguir::
  - Se as operações de abertura para as filas na lista de distribuição todas forem bem-sucedidas ou falharem da mesma maneira, os parâmetros de código de conclusão e de código de razão serão configurados para descrever o resultado comum Os registros de resposta MQRR (se fornecidos pelo aplicativo) não são configurados neste caso.
 

Por exemplo, se cada abertura for bem-sucedida, o código de conclusão será configurado como CCOK e o código de razão será RCNONE. Se cada abertura falhar porque nenhuma das filas existe, os parâmetros serão configurados como CCFAIL e RC2085..
  - Se as operações de abertura para as filas na lista de distribuição não forem todas bem-sucedidas ou falharem da mesma maneira:
    - O parâmetro do código de conclusão é configurado como CCWARN se pelo menos uma abertura for bem-sucedida e como CCFAIL se todos falharem.
    - O parâmetro do código de razão é configurado como RC2136
    - Os registros de resposta (se fornecidos pelo aplicativo) são configurados para os códigos de conclusão individuais e os códigos de razão para as filas na lista de distribuição
- Quando uma lista de distribuição tiver sido aberta com êxito, o identificador *HOBJ* retornado pela chamada poderá ser usado em chamadas MQPUT subsequentes para colocar mensagens em filas na lista de distribuição e em uma chamada MQCLOSE para abrir mão do acesso à lista de distribuição A única opção de fechamento válida para uma lista de distribuição é CONONE..
 

A chamada MQPUT1 também pode ser usada para colocar uma mensagem em uma lista de distribuição; a estrutura MQOD que define as filas na lista é especificada como um parâmetro nessa chamada..
- Cada destino aberto com êxito na lista de distribuição conta como um identificador *separado* ao verificar se o aplicativo excedeu o número máximo permitido de identificadores (consulte o atributo do gerenciador de filas **MaxHandles** ). Isso é verdadeiro mesmo quando dois ou mais dos destinos na lista de distribuição realmente são resolvidos para a mesma fila física Se a chamada MQOPEN



ou MQPUT1 para uma lista de distribuição fizer o número de identificadores em uso pelo aplicativo exceder *MaxHandles*, a chamada falhará com o código de razão RC2017.

- Cada destino aberto com sucesso tem o valor de seu atributo **OpenOutputCount** incrementado em um. Se dois ou mais dos destinos na lista de distribuição realmente forem resolvidos para a mesma fila física, essa fila terá seu atributo **OpenOutputCount** incrementada pelo número de destinos na lista de distribuição que forem resolvidos para essa fila
- Qualquer mudança nas definições de fila que teria feito com que uma manipulação se tornasse inválida se as filas fossem abertas individualmente (por exemplo, uma mudança no caminho de resolução) não faz com que a manipulação da lista de distribuição se torne inválida. No entanto, isso resulta em uma falha para essa fila específica quando a manipulação da lista de distribuições é usada em uma chamada MQPUT subsequente
- É válido para uma lista de distribuição conter apenas um destino.

9. As seguintes notas se aplicam ao uso de filas de clusters.

- Quando uma fila de clusters é aberta pela primeira vez e o gerenciador de filas locais não é um gerenciador de filas de repositório completo, o gerenciador de filas locais obtém informações sobre a fila de clusters de um gerenciador de filas de repositório completo. Quando a rede está ocupada, pode levar vários segundos para que o gerenciador de filas locais receba as informações necessárias do gerenciador de filas do repositório. Como resultado, o aplicativo que emite a chamada MQOPEN pode ter que aguardar até 10 segundos antes que o controle retorne da chamada MQOPEN. Se o gerenciador de fila local não receber as informações necessárias sobre a fila de clusters dentro desse tempo, a chamada falhará com o código de razão RC2189
- Quando uma fila de clusters é aberta e há várias instâncias da fila no cluster, a instância realmente aberta depende das opções especificadas na chamada MQOPEN:

– Se as opções especificadas incluírem uma das seguintes:

- OOBROW
- OOINPQ
- OOINPX
- OOINPS
- OOSSET

a instância da fila de clusters aberta precisa ser a instância local. Se não houver instância local da fila, a chamada MQOPEN falhará.

– Se as opções especificadas não incluírem nenhuma das opções acima, mas incluírem uma ou ambas as opções a seguir:

- OOINQ
- OOOUT

a instância aberta é a instância local se houver uma e uma instância remota, caso contrário. A instância escolhida pelo gerenciador de fila pode, no entanto, ser alterada por uma saída de carga de trabalho do cluster (se houver uma)...

Para obter mais informações sobre filas de clusters, consulte [Filas de clusters](#)..

10. Os aplicativos iniciados por um monitor acionador recebem o nome da fila associada ao aplicativo quando o aplicativo é iniciado. Esse nome da fila pode ser especificado no parâmetro **OBJDSC** para abrir a fila. Consulte a descrição da estrutura do MQTMC para obter detalhes adicionais
11. Ao usar a opção OORLOQ, a fila local já é retornada quando uma fila local, de alias ou modelo é aberta, mas esse não é o caso quando, por exemplo, uma fila remota ou uma fila de cluster não local é aberta; o ResolvedQName e ResolvedQMgrNome são inseridos com o RemoteQName e RemoteQMgrNome localizado na definição de fila remota ou de forma semelhante com a fila de cluster remoto escolhida. Se OORLOQ for especificado ao abrir, por exemplo, uma fila remota, ResolvedQName será a fila de transmissão na qual as mensagens serão colocadas. O Nome ResolvedQMgr será inserido com o nome do gerenciador de filas locais que hospeda a fila de transmissão. Se um usuário estiver autorizado a navegar, entrada ou saída em uma fila, ele terá a

autoridade necessária para especificar esse sinalizador na chamada MQOPEN Nenhuma autoridade especial é necessária.

## Parâmetros

A chamada MQOPEN possui os parâmetros a seguir:

### HCONN (número inteiro assinado de 10 dígitos)-entrada

Manipulação de conexões.

Esta manipulação representa a conexão ao gerenciador de filas. O valor de *HCONN* foi retornado por uma chamada anterior MQCONN ou MQCONNX.

### OBJDSC (MQOD)-entrada/saída

Descritor do objeto

Esta é uma estrutura que identifica o objeto a ser aberto; consulte [“MQOD \(descritor de objeto\) em IBM i”](#) na página 1200 para obter detalhes

Se o campo *ODON* no parâmetro **OBJDSC** for o nome de uma fila modelo, uma fila local dinâmica é criado com os atributos da fila modelo; isso acontece independentemente das opções de abertura especificadas pelo parâmetro **OPTS** Operações subsequentes usando o *HOB*J retornado pela chamada MQOPEN são executadas na nova fila dinâmica e não na fila modelo. Isso é verdadeiro até mesmo para as chamadas MQINQ e MQSET O nome da fila modelo no parâmetro **OBJDSC** é substituído pelo nome da fila dinâmica criada.. O tipo da fila dinâmica é determinado pelo valor do atributo **DefinitionType** da fila modelo (consulte [“Atributos para filas”](#) na página 1416 ).. Para obter informações sobre as opções de fechamento aplicáveis a filas dinâmicas, consulte a descrição da chamada MQCLOSE

### OPTS (número inteiro assinado de 10 dígitos)-entrada

Opções que controlam a ação de MQOPEN..

Pelo menos uma das seguintes opções deve ser especificada:

- OBRW
- OOINP\* (apenas um destes)
- OOINQ
- OOOUT
- OOS
- OORLQ

Outras opções podem ser especificadas conforme necessário.. Se mais de uma opção for necessária, os valores poderão ser incluídos (não inclua a mesma constante mais de uma vez).. As combinações inválidas são observadas; todas as outras combinações são válidas. Apenas opções aplicáveis ao tipo de objeto especificado por *OBJDSC* são permitidas (consulte [Opções MQOPEN válidas para cada tipo de fila](#) ).

**Opções de Acesso:** As opções a seguir controlam o tipo de operações que podem ser executadas no objeto:

#### OOINPQ

Abrir fila para obter mensagens usando padrão definido pela fila.

A fila é aberta para uso com as chamadas MQGET subsequentes O tipo de acesso é compartilhado ou exclusivo, dependendo do valor do atributo da fila **DefInputOpenOption** ; consulte [“Atributos para filas”](#) na página 1416 para obter detalhes.

Esta opção é válida apenas para filas locais, de alias e de modelo; não é válida para filas remotas, listas de distribuições e objetos que não são filas.

## OOINPS

Abrir fila para obter mensagens com acesso compartilhado.

A fila é aberta para uso com as chamadas MQGET subsequentes. A chamada poderá ser bem-sucedida se a fila estiver atualmente aberta por este ou outro aplicativo com OOIINPS, mas falhar com o código de razão RC2042 se a fila estiver atualmente aberta com OOINPX.

Esta opção é válida apenas para filas locais, de alias e de modelo; não é válida para filas remotas, listas de distribuições e objetos que não são filas.

## OOINPX

Abra a fila para obter mensagens com acesso exclusivo.

A fila é aberta para uso com as chamadas MQGET subsequentes. A chamada falha com o código de razão RC2042 se a fila estiver aberta atualmente por este ou outro aplicativo para entrada de qualquer tipo (OOINPS ou OOINPX).

Esta opção é válida apenas para filas locais, de alias e de modelo; não é válida para filas remotas, listas de distribuições e objetos que não são filas.

As notas a seguir se aplicam a essas opções:

- Apenas uma dessas opções pode ser especificada..
- Uma chamada MQOPEN com uma dessas opções pode ser bem-sucedida mesmo se o atributo da fila **InhibitGet** estiver configurado como QAGETI (embora chamadas MQGET subsequentes falharão enquanto o atributo estiver configurado para esse valor).
- Se a fila for definida como não compartilhável (ou seja, o atributo da fila **Shareability** tiver o valor QANSHR), as tentativas de abrir a fila para acesso compartilhado serão tratadas como tentativas de abrir a fila com acesso exclusivo.
- Se uma fila de alias for aberta com uma dessas opções, o teste para uso exclusivo (ou para saber se outro aplicativo possui uso exclusivo) será contra a fila base para a qual o alias é resolvido.
- Essas opções não serão válidas se *ODMN* for o nome de um alias do gerenciador de filas; isso será verdadeiro mesmo se o valor do atributo **RemoteQMGrName** na definição local de uma fila remota usada para alias do gerenciador de filas for o nome do gerenciador de filas locais.

## OOBRW

Abrir fila para procurar mensagens.

A fila é aberta para uso com as chamadas MQGET subsequentes com uma das seguintes opções:

- GMBRWF
- GMBRWN
- GMBRWC

Isso é permitido mesmo se a fila estiver atualmente aberta para OOINPX. Uma chamada MQOPEN com a opção OOBRW estabelece um cursor de procura e posiciona-o logicamente antes da primeira mensagem na fila; consulte o campo *GMOPT* descrito em [“MQGMO \(opções Get-message\) em IBM i”](#) na página 1113 para obter informações adicionais.

Esta opção é válida apenas para filas locais, de alias e de modelo; não é válida para filas remotas, listas de distribuições e objetos que não são filas. Ele também não é válido se *ODMN* for o nome de um alias de gerenciador de filas; isso é verdadeiro mesmo se o valor do atributo **RemoteQMGrName** na definição local de uma fila remota usada para alias de gerenciador de filas for o nome do gerenciador de filas locais.

## OOOUT

Abra a fila para colocar mensagens, ou um tópico ou sequência de tópicos para publicar mensagens.

A fila é aberta para uso com chamadas MQPUT subsequentes.

Uma chamada MQOPEN com essa opção pode ser bem-sucedida mesmo se o atributo da fila **InhibitPut** estiver configurado como QAPUTI (embora chamadas MQPUT subsequentes falharão enquanto o atributo estiver configurado para esse valor).

Esta opção é válida para todos os tipos de fila, incluindo listas de distribuição e tópicos..

### OOINQ

Abrir objeto para consultar atributos.

A fila, a lista de nomes, a definição de processo ou o gerenciador de filas é aberto para uso com chamadas MQINQ subsequentes.

Esta opção é válida para todos os tipos de objetos que não sejam listas de distribuição Não será válido se *ODMN* for o nome de um alias do gerenciador de filas; isso será verdadeiro mesmo se o valor do atributo **RemoteQMgrName** na definição local de uma fila remota usada para alias do gerenciador de filas for o nome do gerenciador de filas locais.

### OOSET

Abra a fila para configurar atributos

A fila é aberta para uso com chamadas MQSET subsequentes.

Esta opção é válida para todos os tipos de filas que não sejam listas de distribuição Não será válido se *ODMN* for o nome de uma definição local de uma fila remota; isso é verdadeiro mesmo se o valor do atributo **RemoteQMgrName** na definição local de uma fila remota usada para alias do gerenciador de filas for o nome do gerenciador de filas locais.

**Opções de ligação:** As opções a seguir se aplicam quando o objeto que está sendo aberto é uma fila de clusters; essas opções controlam a ligação do identificador de filas para uma instância da fila de clusters:

### OOBNDQ

Ligar identificador com destino quando a fila for aberta.

Isso faz com que o gerenciador de filas locais ligue o identificador de filas a uma instância da fila de destino quando a fila é aberta Como resultado, todas as mensagens colocadas usando esse identificador são enviadas para a mesma instância da fila de destino e pela mesma rota

Essa opção é válida somente para filas e afeta somente filas de clusters. Se especificada para uma fila que não seja uma fila de cluster, a opção será ignorada.

### OOBNDN

Não ligar com um destino específico.

Isso para o gerenciador de filas locais que liga a manipulação de filas a uma instância da fila de destino Como resultado, chamadas MQPUT sucessivas usando essa manipulação podem resultar no envio de mensagens para *diferentes* instâncias da fila de destino ou no envio para a mesma instância, mas por rotas diferentes.. Ele também permite que a instância selecionada seja alterada posteriormente pelo gerenciador de filas locais, por um gerenciador de filas remotas ou por um agente de canal de mensagens (MCA), de acordo com as condições de rede...

**Nota:** Os aplicativos cliente e servidor que precisam trocar uma *série* de mensagens para concluir uma transação não devem usar OOBNDN (ou OOBNDQ quando *DefBind* tiver o valor BNDNOT), pois mensagens sucessivas na série podem ser enviadas para diferentes instâncias do aplicativo do servidor.

Se OOBRW ou uma das opções OOINP\* for especificada para uma fila de clusters, o gerenciador de filas será forçado a selecionar a instância local da fila de clusters. Como resultado, a ligação do identificador de filas é corrigida, mesmo se OOBNDN for especificado

Se OOINQ for especificado com OOBNDN, chamadas MQINQ sucessivas usando essa manipulação poderão consultar diferentes instâncias da fila de clusters, embora geralmente todas as instâncias tenham os mesmos valores de atributo.

O OOBNDN é válido apenas para filas e afeta apenas filas de clusters Se especificada para uma fila que não seja uma fila de cluster, a opção será ignorada.

## **OOBNDQ**

Usar ligação padrão para a fila.

Isso faz com que o gerenciador de filas locais ligue o identificador de filas da maneira definida pelo atributo da fila **DefBind** O valor desse atributo é BNDOPN ou BNDNOT

O OOBNDQ será o padrão se OOBNDQ e OOBNDN não forem especificados

O OOBNDQ é definido para auxiliar a documentação do programa.. Não se pretende que essa opção seja usada com qualquer uma das outras duas opções de ligação, mas, como seu valor é zero, esse uso não pode ser detectado.

**Opções de contexto:** As opções a seguir controlam o processamento do contexto da mensagem:

## **OOSAVA**

Salvar o contexto quando a mensagem for recuperada

As informações de contexto estão associadas a este identificador de filas Essas informações são configuradas a partir do contexto de qualquer mensagem recuperada utilizando esse identificador. Para obter mais informações sobre o contexto de mensagem, consulte [Contexto da mensagem](#) e [Informações de contexto de controle](#).

Essas informações de contexto podem ser transmitidas para uma mensagem que é colocada posteriormente em uma fila usando as chamadas MQPUT ou MQPUT1 . Consulte as opções PMPASI e PMPASA descritas em [“MQPMO \(opções Put-message\) em IBM i”](#) na página 1215..

Até que uma mensagem tenha sido recuperada com êxito, o contexto não pode ser transmitido para uma mensagem que está sendo colocada em uma fila

Uma mensagem recuperada usando uma das opções de navegação GMBRW\* não tem suas informações de contexto salvas (embora os campos de contexto no parâmetro **MSGDSC** sejam configurados após uma navegação).

Esta opção é válida apenas para filas locais, de alias e de modelo; não é válida para filas remotas, listas de distribuições e objetos que não são filas. Uma das opções OOINP\* deve ser especificada.

## **OOPASI**

Permitir que contexto de identidade seja transmitido.

Isso permite que a opção PMPASI seja especificada no parâmetro **PMO** quando uma mensagem é colocada em uma fila; isso fornece à mensagem as informações de contexto de identidade de uma fila de entrada que foi aberta com a opção OOSAVA Para obter mais informações sobre o contexto de mensagem, consulte [Contexto da mensagem](#) e [Informações de contexto de controle](#).

A opção OOOOUT deve ser especificada..

Esta opção é válida para todos os tipos da fila, incluindo as listas de distribuição.

## **OOPASA**

Permitir que todo contexto seja transmitido.

Isso permite que a opção PMPASA seja especificada no parâmetro **PMO** quando uma mensagem é colocada em uma fila; isso fornece à mensagem as informações de identidade e de contexto de origem de uma fila de entrada que foi aberta com a opção OOSAVA Para obter mais informações sobre o contexto de mensagem, consulte [Contexto da mensagem](#) e [Informações de contexto de controle](#).

Esta opção implica OOPASI, que não precisa, portanto, ser especificado A opção OOOOUT deve ser especificada..

Esta opção é válida para todos os tipos da fila, incluindo as listas de distribuição.

## **OOSSETI**

Permitir que o contexto de identidade seja definido

Isso permite que a opção PMSETI seja especificada no parâmetro **PMO** quando uma mensagem for colocada em uma fila; isso fornece à mensagem as informações do contexto de identidade contidas no parâmetro **MSGDSC** especificado na chamada MQPUT ou MQPUT1 Para obter mais

informações sobre o contexto de mensagem, consulte [Contexto da mensagem e Informações de contexto de controle](#).

Esta opção implica OOPASI, que não precisa, portanto, ser especificado A opção OOOOUT deve ser especificada..

Esta opção é válida para todos os tipos da fila, incluindo as listas de distribuição.

#### **OOSETA**

Permitir que todo contexto seja configurado.

Isso permite que a opção PMSETA seja especificada no parâmetro **PMO** quando uma mensagem é colocada em uma fila; isso fornece à mensagem as informações de contexto de identidade e de origem contidas no parâmetro **MSGDSC** especificado na chamada MQPUT ou MQPUT1 . Para obter mais informações sobre o contexto de mensagem, consulte [Contexto da mensagem e Informações de contexto de controle](#).

Essa opção implica as seguintes opções, que não precisam, portanto, ser especificadas:

- OOPASI
- OOPASA
- OOSSETI

A opção OOOOUT deve ser especificada..

Esta opção é válida para todos os tipos da fila, incluindo as listas de distribuição.

**Outras opções:** As seguintes opções controlam a verificação de autorização e o que acontece quando o gerenciador de filas está em quiesce:

#### **OOALTU**

Validar com o identificador de usuário especificado.

Isso indica que o campo *ODAU* no parâmetro **OBJDSC** contém um identificador de usuário que deve ser usado para validar essa chamada MQOPEN.. A chamada poderá ser bem-sucedida apenas se esse *ODAU* estiver autorizado a abrir o objeto com as opções de acesso especificadas, independentemente de o identificador de usuário sob o qual o aplicativo está em execução estar autorizado a fazer isso. No entanto, isso não se aplica a nenhuma das opções de contexto especificadas, que são sempre verificadas com relação ao identificador de usuário sob o qual o aplicativo está em execução..

Esta opção é válida para todos os tipos de objeto

#### **OOFIQ**

Falha se o gerenciador de filas estiver quiesce.

Essa opção força a chamada MQOPEN a falhar se o gerenciador de filas estiver em estado quiesce

Esta opção é válida para todos os tipos de objeto

#### **OORLQ**

Insira o nome da fila local que foi aberta

Esta opção especifica que ResolvedQName na estrutura MQOD (se disponível) deve ser inserido com o nome da fila local que foi aberta. O Nome ResolvedQMgrserá inserido de forma semelhante com o nome do gerenciador de filas locais que hospeda a fila local.

<b>Opção</b>	<b>Aliás ( "1" na página 1379 )</b>	<b>Local e Modelo</b>	<b>Remoto</b>	<b>Cluster não local</b>	<b>Lista de distribuição</b>	<b>Tópico</b>
OOINPQ	✓	✓	-	-	-	-

Tabela 750. Opções MQOPEN válidas para cada tipo de fila (continuação)

Opção	Aliás ( “1” na página 1379 )	Local e Modelo	Remoto	Cluster não local	Lista de distribuição	Tópico
OOINPS	✓	✓	-	-	-	-
OOINPX	✓	✓	-	-	-	-
OOBRW	✓	✓	-	-	-	-
OOOUT	✓	✓	✓	✓	✓	✓
OOINQ	✓	✓	“2” na página 1379	✓	-	-
OOSSET	✓	✓	“2” na página 1379	-	-	-
OOBNDQ ( “3” na página 1379 )	✓	✓	✓	✓	✓	-
OOBNDN ( “3” na página 1379 )	✓	✓	✓	✓	✓	-
OOBNDQ ( “3” na página 1379 )	✓	✓	✓	✓	✓	-
OOSAVA	✓	✓	-	-	-	-
OOPASI	✓	✓	✓	✓	✓	“5” na página 1380
OOPASA	✓	✓	✓	✓	✓	“5” na página 1380
OOSSETI	✓	✓	✓	✓	✓	“5” na página 1380
OOSSETA	✓	✓	✓	✓	✓	“5” na página 1380
OOALTU	✓	✓	✓	✓	✓	✓
OOFIQ	✓	✓	✓	✓	✓	✓
OORLQ	✓	✓	✓	✓	-	-

**Notas:**

1. A validade de opções para aliases depende da validade da opção para a fila para a qual o alias é resolvido
2. Esta opção é válida apenas para a definição local de uma fila remota
3. Esta opção pode ser especificada para qualquer tipo de fila, mas será ignorada, se a fila não for uma fila de clusters
4. Este atributo é ignorado para um tópico

- 5. Esses atributos podem ser usados com um tópico, mas afetam apenas o contexto configurado para a mensagem retida, e não os campos de contexto enviados para qualquer assinante

### HOBJ (número inteiro assinado de 10 dígitos)-saída

Manipulação de objetos.

Essa manipulação representa o acesso que foi estabelecido para o objeto Ele deve ser especificado em chamadas de enfileiramento de mensagens subsequentes que operam no objeto Ele deixa de ser válido quando a chamada MQCLOSE é emitida ou quando a unidade de processamento que define o escopo da manipulação é finalizada

O escopo da alça é restrito à menor unidade de processamento paralelo suportado pela plataforma na qual o aplicativo está em execução; o identificador não é válido fora da unidade de processamento paralelo a partir da qual a chamada MQOPEN foi emitida:

- No IBM i, o escopo do identificador é a tarefa que emite a chamada.

### CMPCOD (número inteiro assinado de 10 dígitos)-saída

Código de conclusão.

É um dos seguintes:

#### CCOK

Indica conclusão bem-sucedida.

#### CCWARN

Aviso (conclusão parcial).

#### CCFAIL

A chamada falhou.

### Declaração de RPG

```
C*..1.....2.....3.....4.....5.....6.....7..
C          CALLP      MQOPEN(HCONN : OBJDSC : OPTS :
C                               HOBJ : CMPCOD : REASON)
```

A definição de protótipo para a chamada é:

```
D*..1.....2.....3.....4.....5.....6.....7..
DMQOPEN          PR          EXTPROC('MQOPEN')
D* Connection handle
D HCONN          10I 0 VALUE
D* Object descriptor
D OBJDSC          468A
D* Options that control the action of MQOPEN
D OPTS          10I 0 VALUE
D* Object handle
D HOBJ          10I 0
D* Completion code
D CMPCOD          10I 0
D* Reason code qualifying CMPCOD
D REASON          10I 0
```

### IBM i MQPUT (mensagem Put) em IBM i

A chamada MQPUT coloca uma mensagem em uma fila, lista de distribuição ou em um tópico. A fila, lista de distribuição ou tópico já deve estar aberto.

- [“Sintaxe” na página 1381](#)
- [“Observações de Uso” na página 1381](#)
- [“tópicos” na página 1381](#)



- “MQPUT e MQPUT1” na página 1381
- “Filas de destino” na página 1382
- “Listas de distribuição” na página 1383
- “Cabeçalhos” na página 1384
- “de Saída” na página 1385
- “Parâmetros” na página 1385
- “Declaração de RPG” na página 1390

## Sintaxe

MQPUT (*HCONN, HOBJ, MSGDSC, PMO, BUFLN, BUFFER, CMPCOD, REASON*)

## Observações de Uso

### tópicos

As notas a seguir se aplicam ao uso de tópicos:

1. Ao usar MQPUT para publicar mensagens em um tópico, em que um ou mais assinantes para esse tópico não podem receber a publicação devido a um problema com sua fila de assinantes (por exemplo, ela está cheia), o código de Razão retornado para a chamada MQPUT e o comportamento de entrega depende da configuração dos atributos PMSGDLV ou NPMSGDLV no TOPIC. Observe que a entrega de uma publicação para a fila de mensagens não entregues quando RODLQ é especificado, ou descartar a mensagem quando RODISC é especificado, é considerado uma entrega bem-sucedida da mensagem. Se nenhuma das publicações for entregue, o MQPUT retornará com RC2502 Isso pode ocorrer nos casos a seguir:

- Uma mensagem é publicada para um TOPIC com PMSGDLV ou NPMSGDLV (dependendo da persistência da mensagem) configurado como ALL e qualquer assinatura (durável ou não) tem uma fila que não pode receber a publicação..
- Uma mensagem é publicada para um TOPIC com PMSGDLV ou NPMSGDLV (dependendo da persistência da mensagem) configurado como ALLDUR e uma assinatura durável possui uma fila que não pode receber a publicação

O MQPUT pode retornar com RCNONE, embora as publicações não tenham sido entregues para alguns assinantes nos seguintes casos:

- Uma mensagem é publicada para um TOPIC com PMSGDLV ou NPMSGDLV (dependendo da persistência da mensagem) definido para ALLAVAIL e qualquer assinatura, durável ou não, tem uma fila que não pode receber a publicação.
- Uma mensagem é publicada para um TOPIC com PMSGDLV ou NPMSGDLV (dependendo da persistência da mensagem) configurado para ALLDUR e uma assinatura não durável possui uma fila que não pode receber a publicação.

2. Se não houver assinantes para o tópico sendo usado, a mensagem publicada não será enviada para nenhuma fila e será descartada. Não faz diferença se esta mensagem é persistente ou não persistente, ou se tem expiração ilimitada ou algum tempo de expiração pequeno, ela ainda será descartada se não houver assinantes. A exceção a isso é se a mensagem deve ser retida, nesse caso, embora não seja enviada para nenhuma fila de assinantes, ela é armazenada com relação ao tópico a ser entregue para quaisquer novas assinaturas ou para quaisquer assinantes que solicitam publicações retidas usando MQSUBRQ.

## MQPUT e MQPUT1

As chamadas MQPUT e MQPUT1 podem ser usadas para colocar mensagens em uma fila; a chamada a ser usada depende das circunstâncias

- A chamada MQPUT deve ser usada quando várias mensagens devem ser colocadas na *mesma* fila

Uma chamada MQOPEN especificando a opção OOOUT é emitida primeiro, seguida por uma ou mais solicitações MQPUT para incluir mensagens na fila; finalmente a fila é fechada com uma chamada MQCLOSE. Isso fornece melhor desempenho do que o uso repetido da chamada MQPUT1 ..

- A chamada MQPUT1 deve ser usada quando apenas *uma* mensagem deve ser colocada em uma fila. Essa chamada encapsula as chamadas MQOPEN, MQPUT e MQCLOSE em uma única chamada, minimizando o número de chamadas que devem ser emitidas..

## Filas de destino

Se um aplicativo colocar uma sequência de mensagens na mesma fila sem usar grupos de mensagens, a ordem dessas mensagens é preservada se as seguintes condições forem satisfeitas. Algumas condições se aplicam às filas de destino locais e remotas; outras condições se aplicam apenas às filas de destino remotas

### Condições para filas de destino locais e remotas

- Todas as chamadas MQPUT estão na mesma unidade de trabalho ou nenhuma delas está dentro de uma unidade de trabalho.

Quando mensagens são colocadas em uma fila específica em uma única unidade de trabalho, as mensagens de outros aplicativos podem ser intercaladas com a seqüência de mensagens na fila.

- Todas as chamadas MQPUT são feitas usando o mesmo identificador de objeto *HOBJ*

Em alguns ambientes, a sequência de mensagens também é preservada quando diferentes identificadores de objetos são usados, desde que as chamadas sejam feitas do mesmo aplicativo. O significado de "mesma aplicação" é determinado pelo ambiente:

- No IBM i, o aplicativo é a tarefa
- Todas as mensagens tenham a mesma prioridade.

### Condições adicionais para filas de destino remotas

- Há apenas um caminho do gerenciador da fila de envio para o gerenciador de filas de destino

Se houver uma possibilidade de que algumas mensagens na sequência possam seguir um caminho diferente (por exemplo, devido à reconfiguração, balanceamento de tráfego ou seleção de caminho com base no tamanho da mensagem), a ordem das mensagens no gerenciador de filas de destino não poderá ser garantida.

- As mensagens não são colocadas temporariamente em filas de mensagens não entregues nos gerenciadores de filas de envio, intermediário ou de destino

Se uma ou mais das mensagens forem colocadas temporariamente em uma fila de mensagens não entregues (por exemplo, porque uma fila de transmissão ou a fila de destino está temporariamente cheia), as mensagens poderão chegar na fila de destino fora da sequência...

- As mensagens são persistentes ou não persistentes.

Se um canal na rota entre os gerenciadores de filas de envio e de destino tiver seu atributo **CDNPM** configurado como NPFAS, as mensagens não persistentes poderão saltar adiante das mensagens persistentes, resultando na ordem das mensagens persistentes relativas às mensagens não persistentes que não estão sendo preservadas. No entanto, a ordem de mensagens persistentes relativas entre si e de mensagens não persistentes relativas entre si é preservada.

Se essas condições não forem atendidas, os grupos de mensagens poderão ser usados para preservar a ordem das mensagens, mas observe que isso requer que os aplicativos de envio e de recebimento usem o suporte de agrupamento de mensagem Para obter informações adicionais sobre grupos de mensagens, consulte:

- *MDMFL* Campo no MQMD
- Opção *PMLOGO* em MQPMO
- Opção *GMLOGO* em MQGMO

## Listas de distribuição

As seguintes notas aplicam-se ao uso de listas de distribuição.

1. As mensagens podem ser colocadas em uma lista de distribuição usando uma version-1 ou uma version-2 MQPMO. Se um MQPMO version-1 for usado (ou um MQPMO version-2 com *PMREC* igual a zero), nenhum registro de mensagem put ou registro de resposta poderá ser fornecido pelo aplicativo. Isso significa que não será possível identificar as filas que encontram erros, se a mensagem for enviada com êxito para algumas filas na lista de distribuição e não para outras.

Se os registros de mensagens put ou de resposta forem fornecidos pelo aplicativo, o campo *PMVER* deverá ser configurado como *PMVER2*.

Um MQPMO version-2 também pode ser usado para enviar mensagens para uma única fila que não esteja em uma lista de distribuição, assegurando que *PMREC* seja zero.

2. Os parâmetros de código de conclusão e de código de razão são configurados conforme a seguir::

- Se todas as colocações nas filas na lista de distribuição forem bem-sucedidas ou falharem da mesma maneira, os parâmetros do código de conclusão e do código de razão serão configurados para descrever o resultado comum. Os registros de resposta MQRR (se fornecidos pelo aplicativo) não são configurados neste caso.

Por exemplo, se cada put for bem-sucedido, o código de conclusão será configurado como *CCOK* e o código de razão será *RCNONE*; se cada put falhar porque todas as filas são inibidas para puts, os parâmetros serão configurados como *CCFAIL* e *RC2051*

- Se as colocações nas filas na lista de distribuição não forem bem-sucedidas ou falharem da mesma maneira:
  - O parâmetro de código de conclusão será configurado como *CCWARN* se pelo menos um put for bem-sucedido e como *CCFAIL* se todos falharem.
  - O parâmetro do código de razão é configurado como *RC2136*
  - Os registros de resposta (se fornecidos pelo aplicativo) são configurados para os códigos de conclusão individuais e os códigos de razão para as filas na lista de distribuição

Se o put para um destino falhar porque a abertura para esse destino falhou, os campos no registro de resposta serão configurados como *CCFAIL*, e *RC2137*; esse destino será incluído em *PMIDC*

3. Se um destino na lista de distribuição for resolvido para uma fila local, a mensagem será colocada nessa fila no formato normal (ou seja, não como uma mensagem da lista de distribuições). Se mais de um destino for resolvido para a mesma fila local, uma mensagem será colocada na fila para cada destino.

Se um destino na lista de distribuição for resolvido para uma fila remota, uma mensagem será colocada na fila de transmissão apropriada. Quando vários destinos são resolvidos para a mesma fila de transmissão, uma única mensagem da lista de distribuição contendo esses destinos pode ser colocada na fila de transmissão, mesmo que esses destinos não fossem adjacentes na lista dos destinos fornecidos pelo aplicativo. No entanto, isso poderá ser feito somente se a fila de transmissão suportar mensagens da lista de distribuição (consulte o atributo da fila **DistLists** descrito em [“Atributos para filas”](#) na página 1416).

Se a fila de transmissão não suportar listas de distribuição, uma cópia da mensagem no formato normal será colocada na fila de transmissão para cada destino que usa essa fila de transmissão..

Se uma lista de distribuição com os dados da mensagem do aplicativo for muito grande para uma fila de transmissão, a mensagem da lista de distribuição será dividida em mensagens da lista de distribuição menores, cada uma contendo menos destinos. Se os dados da mensagem do aplicativo apenas se ajustarem na fila, as mensagens da lista de distribuições não poderão ser usadas e o gerenciador de filas gerará uma cópia da mensagem no formato normal para cada destino que usar essa fila de transmissão.

Se destinos diferentes tiverem prioridade ou persistência de mensagem diferente (isso pode ocorrer quando o aplicativo especifica *PRQDEF* ou *PEQDEF*), as mensagens não serão mantidas na mesma mensagem da lista de distribuição. Em vez disso, o gerenciador de filas gera quantas mensagens

de lista de distribuição forem necessárias para acomodar os diferentes valores de prioridade e persistência.

4. Uma colocação em uma lista de distribuição pode resultar em:

- Uma única mensagem de lista de distribuição ou
- Um número de mensagens menores da lista de distribuição ou
- Uma mistura de mensagens da lista de distribuições e mensagens normais ou
- Apenas mensagens normais..

Qual das ocorrências anteriores depende se:

- Os destinos na lista são locais, remotos ou uma mistura.
- Os destinos têm a mesma prioridade e persistência de mensagem.
- As filas de transmissão podem conter mensagens da lista de distribuições
- Os comprimentos máximos de mensagens das filas de transmissão são suficientemente grandes para acomodar a mensagem em forma de lista de distribuições.

No entanto, independentemente de qual das ocorrências acima, cada mensagem *física* resultante (ou seja, cada mensagem normal ou mensagem da lista de distribuição resultante da colocação) conta como apenas *uma* mensagem quando:

- Verificando se o aplicativo excedeu o número máximo permitido de mensagens em uma unidade de trabalho (consulte o atributo do gerenciador de filas **MaxUncommittedMsgs** ).
- Verificando se as condições de acionamento foram atendidas
- Incrementar as profundidades da fila e verificar se a profundidade máxima da fila das filas seria excedida.

5. Qualquer mudança nas definições de fila que teria feito com que uma manipulação se tornasse inválida se as filas fossem abertas individualmente (por exemplo, uma mudança no caminho de resolução) não faz com que a manipulação da lista de distribuição se torne inválida. No entanto, isso resulta em uma falha para essa fila específica quando a manipulação da lista de distribuições é usada em uma chamada MQPUT subsequente

## Cabeçalhos

Se uma mensagem for colocada com uma ou mais estruturas de cabeçalho do IBM MQ no início dos dados da mensagem do aplicativo, o gerenciador de filas executará determinadas verificações nas estruturas de cabeçalho para verificar se elas são válidas. Se o gerenciador de filas detectar um erro, a chamada falhará com um código de razão apropriado. As verificações realizadas variam de acordo com as estruturas específicas presentes. Além disso, as verificações serão executadas apenas se um MQMD version-2 ou posterior for usado na chamada MQPUT ou MQPUT1 ; as verificações não serão executadas se um MQMD version-1 for usado, mesmo se um MQMDE estiver presente no início dos dados da mensagem do aplicativo.

As estruturas do cabeçalho IBM MQ a seguir são validadas completamente pelo gerenciador de filas: MQDH, MQMDE.

Para outras estruturas de cabeçalho IBM MQ , o gerenciador de fila executa alguma validação, mas não verifica todos os campos. As estruturas que não são suportadas pelo gerenciador de filas locais e as estruturas após o primeiro MQDLH na mensagem não são validadas.

Além de verificações gerais nos campos nas estruturas IBM MQ , as condições a seguir devem ser satisfeitas:

- Uma estrutura IBM MQ não deve ser dividida em dois ou mais segmentos-a estrutura deve estar totalmente contida dentro de um segmento
- A soma dos comprimentos das estruturas em uma mensagem PCF deve ser igual ao comprimento especificado pelo parâmetro **BUFLen** na chamada MQPUT ou MQPUT1 . Uma mensagem PCF é uma mensagem que possui um dos seguintes nomes de formato:

- FMADMN
- FMEVNT
- FMPCF
- As estruturas IBM MQ não devem ser truncadas, exceto nas seguintes situações em que estruturas truncadas são permitidas:
  - Mensagens que são mensagens de relatório..
  - Mensagens PCF..
  - Mensagens contendo uma estrutura MQDLH. (As estruturas *após* o primeiro MQDLH podem ser truncadas; as estruturas que precedem o MQDLH não podem.)

## de Saída

O parâmetro **BUFFER** mostrado no exemplo de programação de RPG é declarado como uma cadeia; isso restringe o comprimento máximo do parâmetro a 256 bytes. Se um buffer maior for necessário, o parâmetro deverá ser declarado como uma estrutura ou como um campo em um arquivo físico. Isso irá aumentar o comprimento máximo possível para aproximadamente 32 KB.

## Parâmetros

A chamada MQPUT tem os parâmetros a seguir:

### HCONN (número inteiro assinado de 10 dígitos)-entrada

Manipulação de conexões.

Esta manipulação representa a conexão ao gerenciador de filas. O valor de *HCONN* foi retornado por uma chamada anterior MQCONN ou MQCONNX.

### HOBJ (número inteiro assinado de 10 dígitos)-entrada

Manipulação de objetos.

Esse identificador representa a fila na qual a mensagem é incluída, ou o tópico no qual a mensagem é publicada. O valor de *HOBJ* foi retornado por uma chamada de MQOPEN anterior que especificou a opção OOOUT.

### MSGDSC (MQMD)-entrada/saída

Descritor de mensagens

Esta estrutura descreve os atributos da mensagem que está sendo enviada e recebe informações sobre a mensagem após o pedido de colocação ser concluído. Consulte [“MQMD \(Descritor de mensagens\) em IBM i”](#) na página 1148 para obter detalhes.

Se o aplicativo fornecer um MQMD version-1, os dados da mensagem poderão ser prefixados com uma estrutura MQMDE para especificar valores para os campos que existem no MQMD version-2, mas não no version-1. O campo *MDFMT* no MQMD deve estar configurado como FMMDE, para indicar que um MQMDE está presente. Consulte [“MQMDE \(extensão do descritor de mensagens\) em IBM i”](#) na página 1194 para obter mais detalhes.

### PMO (MQPMO)-entrada/saída

Opções que controlam a ação de MQPUT..

Consulte [“MQPMO \(opções Put-message\) em IBM i”](#) na página 1215 para obter detalhes.

### BUFLen (número inteiro assinado de 10 dígitos)-entrada

Comprimento da mensagem em *BUFFER*..

Zero é válido e indica que a mensagem não contém nenhum dado do aplicativo. O limite superior para *BUFLen* depende de vários fatores:

- Se a fila de destino for uma fila compartilhada, o limite superior será 63 KB (64 512 bytes).

- Se o destino for uma fila local ou for resolvido para uma fila local (mas não for uma fila compartilhada), o limite superior dependerá se:
  - O gerenciador de filas locais suporta a segmentação
  - O aplicativo de envio especifica a sinalização que permite que o gerenciador de filas segmente a mensagem. Esse sinalizador é MFSEGA e pode ser especificado em um MQMD version-2 ou em um MQMDE usado com um MQMD version-1 .

Se ambas as condições forem satisfeitas, *BUFLEN* não poderá exceder 999 999 999 menos o valor do campo *MDOFF* no MQMD. Portanto, a mensagem lógica mais longa que pode ser colocada é 999 999 999 bytes (quando *MDOFF* é zero). No entanto, as restrições de recursos impostas pelo sistema operacional ou ambiente no qual o aplicativo está em execução podem resultar em um limite inferior

Se uma ou ambas as condições descritas anteriormente não forem satisfeitas, o *BUFLEN* não poderá exceder o menor atributo **MaxMsgLength** da fila e o atributo **MaxMsgLength** do gerenciador de filas.

- Se o destino for uma fila remota ou for resolvido para uma fila remota, as condições para filas locais se aplicam, *mas em cada gerenciador de filas por meio do qual a mensagem deve passar para atingir a fila de destino* ; em especial:
  1. A fila de transmissão local usada para armazenar a mensagem temporariamente no gerenciador de filas locais
  2. Filas de transmissão intermediárias (se houver) utilizadas para armazenar a mensagem nos gerenciadores de fila na rota entre os gerenciadores de filas local e de destino
  3. A fila de destino no gerenciador de filas de destino

A mensagem mais longa que pode ser colocada é, portanto, controlada pelo mais restritivo dessas filas e gerenciadores de filas.

Quando uma mensagem está em uma fila de transmissão, informações adicionais residem nos dados da mensagem e isso reduz a quantidade de dados do aplicativo que podem ser transportados. Nessa situação, recomenda-se que os bytes LNMHD sejam subtraídos dos valores *MaxMsgLength* das filas de transmissão ao determinar o limite para *BUFLEN*.

**Nota:** Apenas a falha em obedecer à condição 1 pode ser diagnosticada de forma síncrona (com o código de razão RC2030 ou RC2031) quando a mensagem é colocada. Se as condições 2 ou 3 não forem satisfeitas, a mensagem será redirecionada para uma fila de mensagens não entregues (não entregues), em um gerenciador de filas intermediário ou no gerenciador de filas de destino. Se isso acontecer, uma mensagem de relatório será gerada se uma foi solicitada pelo emissor.

### **BUFFER (cadeia de bits de 1 byte x BUFLEN)-entrada**

Os dados da mensagem.

Este é um buffer que contém os dados do aplicativo a serem enviados. O buffer deve ser alinhado em um limite apropriado para a natureza dos dados na mensagem. O alinhamento de 4 bytes deve ser adequado para a maioria das mensagens (incluindo mensagens contendo estruturas de cabeçalho MQ), mas algumas mensagens podem requerer alinhamento mais rigoroso. Por exemplo, uma mensagem contendo um número inteiro binário de 64 bits pode requerer alinhamento de 8 bytes.

Se *BUFFER* contiver dados de caractere, dados numéricos ou ambos, os campos *MDCSI* e *MDENC* no parâmetro **MSGDSC** deverão ser configurados para os valores apropriados para os dados; isso permitirá que o receptor da mensagem converta os dados (se necessário) para o conjunto de caracteres e a codificação usados pelo receptor..

**Nota:** Todos os outros parâmetros na chamada MQPUT devem estar no conjunto de caracteres fornecido pelo atributo do gerenciador de fila **CodedCharSetId**, e na codificação do gerenciador de filas local fornecido pelo ENNAT

## **CMPCOD (número inteiro assinado de 10 dígitos)-saída**

Código de conclusão.

É um dos seguintes:

### **CCOK**

Indica conclusão bem-sucedida.

### **CCWARN**

Aviso (conclusão parcial).

### **CCFAIL**

A chamada falhou.

## **REASON (número inteiro assinado de 10 dígitos)-saída**

Código de razão qualificando *CMPCOD*.

Se *CMPCOD* for CCOK:

### **RCNONE**

(0, X'000') Nenhuma razão a relatar.

Se *CMPCOD* for CCWARN:

### **RC2104**

(2104, X'838 ') Opção de relatório no descritor da mensagem não reconhecida.

### **RC2136**

(2136, X'858 ') Códigos de razão múltiplos retornados.

Se *CMPCOD* for CCFAIL:

### **RC2004**

(2004, X'7D4') Parâmetro de buffer inválido.

### **RC2005**

(2005, X'7D5') Parâmetro de comprimento de buffer inválido.

### **RC2009**

(2009, X'7D9') Conexão ao gerenciador de filas perdida.

### **RC2013**

(2013, X'7DD') Tempo de expiração inválido.

### **RC2014**

(2014, X'7DE') Código de feedback inválido.

### **RC2018**

(2018, X'7E2') Manipulação de conexões não válida.

### **RC2019**

(2019, X'7E3') Manipulação de objetos não válida.

### **RC2024**

(2024, X'7E8') Nenhuma mensagem adicional pode ser manipulada dentro da unidade de trabalho atual.

### **RC2026**

(2026, X'7EA') Descritor de mensagem inválido.

### **RC2027**

(2027, X'7EB') Fila de resposta ausente.

### **RC2029**

(2029, X'7ED') Tipo de mensagem no descritor de mensagens inválido.

### **RC2030**

(2030, X'7EE') Comprimento da mensagem maior que o máximo para a fila.

### **RC2031**

(2031, X'7EF') Comprimento da mensagem maior que o máximo para o gerenciador de filas.

**RC2039**

(2039, X'7F7') Fila não aberta para saída.

**RC2041**

(2041, X'7F9') Definição de objeto alterada desde a abertura.

**RC2046**

(2046, X'7FE') Opções não válidas ou não consistentes.

**RC2047**

(2047, X'7FF') Persistência inválida.

**RC2048**

(2048, X'800 ') A Fila não suporta mensagens persistentes.

**RC2050**

(2050, X'802 ') A prioridade da mensagem não é válida.

**RC2051**

(2051, X'803 ') Chamadas de put inibidas para a fila

**RC2052**

(2052, X'804') A fila foi excluída.

**RC2053**

(2053, X'805 ') A fila já contém número máximo de mensagens.

**RC2056**

(2056, X'808 ') Nenhum espaço disponível no disco para fila.

**RC2058**

(2058, X'80A') Nome do gerenciador de filas não válido ou não conhecido.

**RC2059**

(2059, X'80B') Gerenciador de filas não disponível para conexão.

**RC2061**

(2061, X'80D') Opções de relatório no descritor de mensagem não válidas.

**RC2071**

(2071, X'817') Armazenamento insuficiente disponível.

**RC2072**

(2072, X'818 ') Suporte de ponto de sincronização não disponível.

**RC2093**

(2093, X'82D') Fila não aberta para passar todo o contexto.

**RC2094**

(2094, X'82E') Fila não aberta para transmitir contexto de identidade.

**RC2095**

(2095, X'82F') Fila não aberta para configurar todo o contexto.

**RC2096**

(2096, X'830 ') Fila não aberta para o contexto de identidade configurado.

**RC2097**

(2097, X'831 ') O identificador de fila referido não salva o contexto..

**RC2098**

(2098, X'832 ') Contexto não disponível para o identificador de fila referido.

**RC2101**

(2101, X'835') Objeto danificado.

**RC2102**

(2102, X'836') Recursos insuficientes do sistema disponíveis.

**RC2135**

(2135, X'857 ') Estrutura do cabeçalho de distribuição inválida.

**RC2136**

(2136, X'858 ') Códigos de razão múltiplos retornados.



- RC2137**  
(2137, X'859') Objeto não aberto com sucesso.
- RC2149**  
(2149, X'865 ') estruturas PCF inválidas.
- RC2154**  
(2154, X'86A') Número de registros presentes não válidos.
- RC2156**  
(2156, X'86C') Registros de resposta inválidos.
- RC2158**  
(2158, X'86E') Sinalizadores de registro de mensagem de colocação não válidos.
- RC2159**  
(2159, X'86F') Os registros de mensagens Put não são válidos.
- RC2161**  
(2161, X'871') Quiesce do gerenciador de filas.
- RC2162**  
(2162, X'872') Gerenciador de filas sendo encerrado.
- RC2173**  
(2173, X'87D') Estrutura de opções de mensagem de entrada inválida.
- RC2185**  
(2185, X'889 ') Especificação de persistência inconsistente.
- RC2188**  
(2188, X'88C') Chamada rejeitada pela saída de carga de trabalho do cluster.
- RC2189**  
(2189, X'88D') Falha na resolução do nome do cluster.
- RC2195**  
(2195, X'893') Ocorreu um erro inesperado.
- RC2219**  
(2219, X'8AB') chamada MQI reinserida antes da chamada anterior ser concluída.
- RC2241**  
(2241, X'8C1') Grupo de mensagens não concluído.
- RC2242**  
(2242, X'8C2') Mensagem lógica não concluída.
- RC2245**  
(2245, X'8C5') Especificação de unidade de trabalho inconsistente.
- RC2248**  
(2248, X'8C8') Extensão do descritor de mensagens inválida.
- RC2249**  
(2249, X'8C9') Sinalizadores de mensagem não válidos.
- RC2250**  
(2250, X'8CA') Número de sequência da mensagem inválido.
- RC2251**  
(2251, X'8CB') O deslocamento do segmento da mensagem não é válido..
- RC2252**  
(2252, X'8CC') Comprimento original inválido.
- RC2253**  
(2253, X'8CD') O comprimento dos dados no segmento de mensagem é zero...
- RC2255**  
(2255, X'8CF') Unidade de trabalho não disponível para o gerenciador de filas a usar.
- RC2257**  
(2257, X'8D1') Versão errada do MQMD fornecido.

**RC2258**

(2258, X'8D2') Identificador de grupo inválido.

**RC2266**

(2266, X'8DA') Falha na saída de carga de trabalho do cluster.

**RC2269**

(2269, X'8DD') Erro de recurso de cluster

**RC2270**

(2270, X'8DE') Nenhuma fila de destinos disponível.

**RC2420**

(2420) Uma chamada MQPUT foi emitida, mas os dados da mensagem contêm uma estrutura MQEPH que não é válida..

**RC2479**

(2479, X'9AF') Publicação não pôde ser retida.

**RC2480**

(2480, X'9B0') O tipo de destino foi alterado: a fila de alias se referia a uma fila, mas agora se refere a um tópico.

**RC2502**

(2502, X'9C6') A publicação falhou e a publicação não foi entregue a nenhum assinante

**RC2551**

(2551, X'9F7') A sequência de seleção especificada não está disponível.

**RC2554**

(2554, X'9FA') O conteúdo da mensagem não pôde ser analisado para determinar se a mensagem deve ser entregue a um assinante com um seletor de mensagens estendidas...

**Declaração de RPG**

```

C*.1.....2.....3.....4.....5.....6.....7..
C          CALLP      MQPUT(HCONN : HOBJ : MSGDSC : PMO :
C                               BUFLLEN : BUFFER : CMPCOD :
C                               REASON)

```

A definição de protótipo para a chamada é:

```

D*.1.....2.....3.....4.....5.....6.....7..
DMQPUT      PR          EXTPROC('MQPUT')
D* Connection handle
D HCONN          10I 0 VALUE
D* Object handle
D HOBJ          10I 0 VALUE
D* Message descriptor
D MSGDSC          364A
D* Options that control the action of MQPUT
D PMO          200A
D* Length of the message in Buffer
D BUFLLEN          10I 0 VALUE
D* Message data
D BUFFER          * VALUE
D* Completion code
D CMPCOD          10I 0
D* Reason code qualifying CMPCOD
D REASON          10I 0

```

**IBM i MQPUT1 (Colocar uma mensagem) em IBM i .**

A chamada MQPUT1 coloca uma mensagem em uma fila ou lista de distribuição ou em um tópico. A fila, lista de distribuição ou tópico não precisa ser aberto.

- [“Sintaxe” na página 1391](#)

- [“Observações de Uso” na página 1391](#)
- [“Parâmetros” na página 1392](#)
- [“Declaração de RPG” na página 1396](#)

## Sintaxe

MQPUT1 (*HCONN, OBJDSC, MSGDSC, PMO, BUFLen, BUFFER, CMPCOD, REASON*)

## Observações de Uso

1. As chamadas MQPUT e MQPUT1 podem ser usadas para colocar mensagens em uma fila; a chamada a ser usada depende das circunstâncias:
  - A chamada MQPUT deve ser usada quando várias mensagens devem ser colocadas na *mesma* fila. Uma chamada MQOPEN especificando a opção OOOOUT é emitida primeiro, seguida por uma ou mais solicitações MQPUT para incluir mensagens na fila; finalmente a fila é fechada com uma chamada MQCLOSE. Isso fornece melhor desempenho do que o uso repetido da chamada MQPUT1 ..
  - A chamada MQPUT1 deve ser usada quando apenas *uma* mensagem deve ser colocada em uma fila. Essa chamada encapsula as chamadas MQOPEN, MQPUT e MQCLOSE em uma única chamada, minimizando o número de chamadas que devem ser emitidas..
2. Se um aplicativo colocar uma sequência de mensagens na mesma fila sem usar grupos de mensagens, a ordem dessas mensagens será preservada desde que determinadas condições sejam satisfeitas. No entanto, na maioria dos ambientes, a chamada MQPUT1 não satisfaz essas condições e, portanto, não preserva a ordem de mensagens. A chamada MQPUT deve ser usada nesses ambientes. Consulte as notas de uso na descrição da chamada MQPUT para obter detalhes.
3. A chamada MQPUT1 pode ser usada para colocar mensagens em listas de distribuição. Para obter informações gerais sobre isso, consulte as notas de uso das chamadas MQOPEN e MQPUT.

As diferenças a seguir se aplicam ao usar a chamada MQPUT1 :

- a. Se os registros de resposta MQRR forem fornecidos pelo aplicativo, eles deverão ser fornecidos usando a estrutura MQOD; não poderão ser fornecidos usando a estrutura MQPMO.
  - b. O código de razão RC2137 nunca é retornado por MQPUT1 nos registros de resposta; se uma fila falhar ao abrir, o registro de resposta para essa fila contém o código de razão real resultante da operação de abertura.
 

Se uma operação de abertura para uma fila for bem-sucedida com um código de conclusão de CCWARN, o código de conclusão e o código de razão no registro de resposta para essa fila serão substituídos pelos códigos de conclusão e de razão resultantes da operação de colocação

Como com as chamadas MQOPEN e MQPUT, o gerenciador de filas configura os registros de resposta (se fornecidos) somente quando o resultado da chamada não é o mesmo para todas as filas na lista de distribuição; isso é indicado pela chamada concluindo com o código de razão RC2136.
4. Se a chamada MQPUT1 for usada para colocar uma mensagem em uma fila de clusters, a chamada se comporta como se OOBNDN tivesse sido especificado na chamada MQOPEN.
  5. Se uma mensagem for colocada com uma ou mais estruturas de cabeçalho do IBM MQ no início dos dados da mensagem do aplicativo, o gerenciador de filas executará determinadas verificações nas estruturas de cabeçalho para verificar se elas são válidas. Para obter mais informações sobre isso, consulte as notas de uso para a chamada MQPUT
  6. Se mais de uma das situações de aviso surgir (consulte o parâmetro **CMPCOD** ), o código de razão retornado será o *primeiro* na lista a seguir que se aplica:
    - a. RC2136
    - b. RC2242
    - c. RC2241

d. RC2049 ou RC2104

7. O parâmetro **BUFFER** mostrado no exemplo de programação de RPG é declarado como uma cadeia; isso restringe o comprimento máximo do parâmetro a 256 bytes. Se um buffer maior for necessário, o parâmetro deverá ser declarado como uma estrutura ou como um campo em um arquivo físico. Isso irá aumentar o comprimento máximo possível para aproximadamente 32 KB.

## Parâmetros

A chamada MQPUT1 tem os seguintes parâmetros:

### **HCONN (número inteiro assinado de 10 dígitos)-entrada**

Manipulação de conexões.

Esta manipulação representa a conexão ao gerenciador de filas. O valor de *HCONN* foi retornado por uma chamada anterior MQCONN ou MQCONNX.

### **OBJDSC (MQOD)-entrada/saída**

Descritor do objeto

Esta é uma estrutura que identifica a fila na qual a mensagem é adicionada. Consulte [“MQOD \(descritor de objeto\) em IBM i”](#) na página 1200 para obter detalhes.

O usuário deve estar autorizado a abrir a fila para a saída. A fila **não** deve ser uma fila modelo..

### **MSGDSC (MQMD)-entrada/saída**

Descritor de mensagens

Essa estrutura descreve os atributos da mensagem que está sendo enviada e recebe informações de feedback após a conclusão da solicitação de colocação. Consulte [“MQMD \(Descritor de mensagens\) em IBM i”](#) na página 1148 para obter detalhes.

Se o aplicativo fornecer um MQMD version-1, os dados da mensagem poderão ser prefixados com uma estrutura MQMDE para especificar valores para os campos que existem no MQMD version-2, mas não no version-1. O campo *MDFMT* no MQMD deve estar configurado como FMMDE, para indicar que um MQMDE está presente. Consulte [“MQMDE \(extensão do descritor de mensagens\) em IBM i”](#) na página 1194 para obter mais detalhes.

### **PMO (MQPMO)-entrada/saída**

Opções que controlam a ação de MQPUT1.

Consulte [“MQPMO \(opções Put-message\) em IBM i”](#) na página 1215 para obter detalhes.

### **BUFLEN (número inteiro assinado de 10 dígitos)-entrada**

Comprimento da mensagem em *BUFFER*..

Zero é válido e indica que a mensagem não contém nenhum dado do aplicativo. O limite superior depende de vários fatores; consulte a descrição do parâmetro **BUFLEN** da chamada MQPUT para obter detalhes adicionais.

### **BUFFER (cadeia de bits de 1 byte x BUFLEN)-entrada**

Os dados da mensagem.

Este é um buffer que contém os dados da mensagem do aplicativo a serem enviados. O buffer deve ser alinhado em um limite apropriado para a natureza dos dados na mensagem. O alinhamento de 4 bytes deve ser adequado para a maioria das mensagens (incluindo mensagens contendo estruturas de cabeçalho IBM MQ), mas algumas mensagens podem requerer alinhamento mais rigoroso. Por exemplo, uma mensagem contendo um número inteiro binário de 64 bits pode requerer alinhamento de 8 bytes.

Se *BUFFER* contiver dados de caractere, dados numéricos ou ambos, os campos *MDCSI* e *MDENC* no parâmetro **MSGDSC** deverão ser configurados para os valores apropriados para os dados; isso

permitirá que o receptor da mensagem converta os dados (se necessário) para o conjunto de caracteres e a codificação usados pelo receptor..

**Nota:** Todos os outros parâmetros na chamada MQPUT1 devem estar no conjunto de caracteres fornecido pelo atributo do gerenciador de fila **CodedCharSetId** e a codificação do gerenciador de filas local fornecido por ENNAT...

### **CMPCOD (número inteiro assinado de 10 dígitos)-saída**

Código de conclusão.

É um dos seguintes:

#### **CCOK**

Indica conclusão bem-sucedida.

#### **CCWARN**

Aviso (conclusão parcial).

#### **CCFAIL**

A chamada falhou.

### **REASON (número inteiro assinado de 10 dígitos)-saída**

Código de razão qualificando *CMPCOD*.

Se *CMPCOD* for CCOK:

#### **RCNONE**

(0, X'000') Nenhuma razão a relatar.

Se *CMPCOD* for CCWARN:

#### **RC2104**

(2104, X'838 ') Opção de relatório no descritor da mensagem não reconhecida.

#### **RC2136**

(2136, X'858 ') Códigos de razão múltiplos retornados.

#### **RC2049**

(2049, X'801 ') A prioridade da mensagem excede o valor máximo suportado.

#### **RC2241**

(2241, X'8C1') Grupo de mensagens não concluído.

#### **RC2242**

(2242, X'8C2') Mensagem lógica não concluída.

Se *CMPCOD* for CCFAIL:

#### **RC2001**

(2001, X'7D1') A fila base do alias não é um tipo válido.

#### **RC2004**

(2004, X'7D4') Parâmetro de buffer inválido.

#### **RC2005**

(2005, X'7D5') Parâmetro de comprimento de buffer inválido.

#### **RC2009**

(2009, X'7D9') Conexão ao gerenciador de filas perdida.

#### **RC2013**

(2013, X'7DD') Tempo de expiração inválido.

#### **RC2014**

(2014, X'7DE') Código de feedback inválido.

#### **RC2017**

(2017, X'7E1') Não há mais identificadores disponíveis

#### **RC2018**

(2018, X'7E2') Manipulação de conexões não válida.

**RC2024**

(2024, X'7E8') Nenhuma mensagem adicional pode ser manipulada dentro da unidade de trabalho atual.

**RC2026**

(2026, X'7EA') Descritor de mensagem inválido.

**RC2027**

(2027, X'7EB') Fila de resposta ausente.

**RC2029**

(2029, X'7ED') Tipo de mensagem no descritor de mensagens inválido.

**RC2030**

(2030, X'7EE') Comprimento da mensagem maior que o máximo para a fila.

**RC2031**

(2031, X'7EF') Comprimento da mensagem maior que o máximo para o gerenciador de filas.

**RC2035**

(2035, X'7F3') Não autorizado para acesso.

**RC2042**

(2042, X'7FA') Objeto já aberto com opções conflitantes.

**RC2043**

(2043, X'7FB') Tipo de objeto inválido.

**RC2044**

(2044, X'7FC') Estrutura do descritor de objeto inválida.

**RC2046**

(2046, X'7FE') Opções não válidas ou não consistentes.

**RC2047**

(2047, X'7FF') Persistência inválida.

**RC2048**

(2048, X'800 ') A Fila não suporta mensagens persistentes.

**RC2050**

(2050, X'802 ') A prioridade da mensagem não é válida.

**RC2051**

(2051, X'803 ') Chamadas de put inibidas para a fila

**RC2052**

(2052, X'804') A fila foi excluída.

**RC2053**

(2053, X'805 ') A fila já contém número máximo de mensagens.

**RC2056**

(2056, X'808 ') Nenhum espaço disponível no disco para fila.

**RC2057**

(2057, X'809 ') Tipo de fila inválido.

**RC2058**

(2058, X'80A') Nome do gerenciador de filas não válido ou não conhecido.

**RC2059**

(2059, X'80B') Gerenciador de filas não disponível para conexão.

**RC2061**

(2061, X'80D') Opções de relatório no descritor de mensagem não válidas.

**RC2063**

(2063, X'80F') Ocorreu um erro na segurança.

**RC2071**

(2071, X'817') Armazenamento insuficiente disponível.

- RC2072**  
(2072, X'818 ') Suporte de ponto de sincronização não disponível.
- RC2082**  
(2082, X'822 ') Fila base de alias desconhecida.
- RC2085**  
(2085, X'825 ') Nome do objeto desconhecido.
- RC2086**  
(2086, X'826 ') Gerenciador de filas de objeto desconhecido.
- RC2087**  
(2087, X'827 ') Gerenciador de filas remotas desconhecido.
- RC2091**  
(2091, X'82B') Fila de transmissão não local.
- RC2092**  
(2092, X'82C') Fila de transmissão com uso errado.
- RC2097**  
(2097, X'831 ') O identificador de fila referido não salva o contexto..
- RC2098**  
(2098, X'832 ') Contexto não disponível para o identificador de fila referido.
- RC2101**  
(2101, X'835') Objeto danificado.
- RC2102**  
(2102, X'836') Recursos insuficientes do sistema disponíveis.
- RC2135**  
(2135, X'857 ') Estrutura do cabeçalho de distribuição inválida.
- RC2136**  
(2136, X'858 ') Códigos de razão múltiplos retornados.
- RC2149**  
(2149, X'865 ') estruturas PCF inválidas.
- RC2154**  
(2154, X'86A') Número de registros presentes não válidos.
- RC2155**  
(2155, X'86B') Registros de objeto inválidos.
- RC2156**  
(2156, X'86C') Registros de resposta inválidos.
- RC2158**  
(2158, X'86E') Sinalizadores de registro de mensagem de colocação não válidos.
- RC2159**  
(2159, X'86F') Os registros de mensagens Put não são válidos.
- RC2161**  
(2161, X'871') Quiesce do gerenciador de filas.
- RC2162**  
(2162, X'872') Gerenciador de filas sendo encerrado.
- RC2173**  
(2173, X'87D') Estrutura de opções de mensagem de entrada inválida.
- RC2184**  
(2184, X'888 ') Nome da fila remota inválido.
- RC2188**  
(2188, X'88C') Chamada rejeitada pela saída de carga de trabalho do cluster.
- RC2189**  
(2189, X'88D') Falha na resolução do nome do cluster.

**RC2195**

(2195, X'893') Ocorreu um erro inesperado.

**RC2196**

(2196, X'894 ') Fila de transmissão desconhecida.

**RC2197**

(2197, X'895 ') Fila de transmissão padrão desconhecida.

**RC2198**

(2198, X'896 ') Fila de transmissão padrão não local.

**RC2199**

(2199, X'897 ') Erro de uso da fila de transmissão padrão

**RC2258**

(2258, X'8D2') Identificador de grupo inválido.

**RC2248**

(2248, X'8C8') Extensão do descritor de mensagens inválida.

**RC2219**

(2219, X'8AB') chamada MQI reinserida antes da chamada anterior ser concluída.

**RC2249**

(2249, X'8C9') Sinalizadores de mensagem não válidos.

**RC2250**

(2250, X'8CA') Número de sequência da mensagem inválido.

**RC2251**

(2251, X'8CB') O deslocamento do segmento da mensagem não é válido..

**RC2252**

(2252, X'8CC') Comprimento original inválido.

**RC2253**

(2253, X'8CD') O comprimento dos dados no segmento de mensagem é zero...

**RC2255**

(2255, X'8CF') Unidade de trabalho não disponível para o gerenciador de filas a usar.

**RC2257**

(2257, X'8D1') Versão errada do MQMD fornecido.

**RC2266**

(2266, X'8DA') Falha na saída de carga de trabalho do cluster.

**RC2269**

(2269, X'8DD') Erro de recurso de cluster

**RC2270**

(2270, X'8DE') Nenhuma fila de destinos disponível.

**RC2420**

(2420) Uma chamada MQPUT1 foi emitida, mas os dados da mensagem contêm uma estrutura MQEPH que não é válida.

**RC2551**

(2551, X'9F7') A sequência de seleção especificada não está disponível.

**RC2554**

(2554, X'9FA') O conteúdo da mensagem não pôde ser analisado para determinar se a mensagem deve ser entregue a um assinante com um seletor de mensagens estendidas...

**Declaração de RPG**

```

C*..1.....2.....3.....4.....5.....6.....7..
C          CALLP      MQPUT1(HCONN : OBJDSC : MSGDSC :

```



C  
C

PMO : BUFLN : BUFFER :  
CMPCOD : REASON)

A definição de protótipo para a chamada é:

```
D*.1.....2.....3.....4.....5.....6.....7..
DMQPUT1      PR      EXTPROC('MQPUT1')
D* Connection handle
D HCONN      10I 0 VALUE
D* Object descriptor
D OBJDSC     468A
D* Message descriptor
D MSGDSC     364A
D* Options that control the action of MQPUT1
D PMO        200A
D* Length of the message in BUFFER
D BUFLN      10I 0 VALUE
D* Message data
D BUFFER     * VALUE
D* Completion code
D CMPCOD     10I 0
D* Reason code qualifying CMPCOD
D REASON     10I 0
```

IBM i

## MQSET (Configurar atributos do objeto) no IBM i

A chamada MQSET é usada para alterar os atributos de um objeto representado por um identificador O objeto deve ser uma fila..

- [“Sintaxe” na página 1397](#)
- [“Observações de Uso” na página 1397](#)
- [“Parâmetros” na página 1398](#)
- [“Declaração de RPG” na página 1401](#)

### Sintaxe

MQSET (*HCONN, HOBJ, SELCNT, SELS, IACNT, INTATR, CALEN, CHRATR, CMPCOD, REASON*)

### Observações de Uso

1. Usando esta chamada, o aplicativo pode especificar uma matriz de atributos de número inteiro ou uma coleção de sequências de atributos de caracteres ou ambos. Se não ocorrerem erros, os atributos especificados serão todos configurados simultaneamente. Se ocorrer um erro (por exemplo, se um seletor não for válido ou for feita uma tentativa de configurar um atributo para um valor inválido), a chamada falhará e nenhum atributo será configurado.
2. Os valores de atributos podem ser determinados usando a chamada MQINQ; consulte [“MQINQ \(Consultar sobre atributos do objeto\) em IBM i” na página 1352](#) para obter detalhes.  
**Nota:** Nem todos os atributos com valores que podem ser consultados usando a chamada MQINQ podem ter seus valores alterados usando a chamada MQSET. Por exemplo, nenhum atributo do objeto de processo ou do gerenciador de fila pode ser configurado com essa chamada.
3. As mudanças de atributo são preservadas em reinicializações do gerenciador de filas (além de alterações em filas dinâmicas temporárias, que não sobrevivem às reinicializações do gerenciador de filas)..
4. Não é possível alterar os atributos de uma fila de modelo usando a chamada MQSET. No entanto, se você abrir uma fila modelo usando a chamada MQOPEN com a opção MQOO\_SET, será possível usar a chamada MQSET para configurar os atributos da fila local dinâmica criada pela chamada MQOPEN.
5. Se o objeto que está sendo configurado for uma fila de clusters, deve haver uma instância local da fila de clusters para que a abertura seja bem-sucedida.

Para obter mais informações sobre atributos de objeto, consulte:

- [“Atributos para filas” na página 1416](#)
- [“Atributos para Listas de Nomes” na página 1446](#)
- [“Atributos para definições de processo em IBM i” na página 1447](#)
- [“Atributos para o gerenciador de filas no IBM i” na página 1449](#)

## Parâmetros

A chamada MQSET possui os parâmetros a seguir:

### **HCONN (número inteiro assinado de 10 dígitos)-entrada**

Manipulação de conexões.

Esta manipulação representa a conexão ao gerenciador de filas. O valor de HCONN foi retornado por uma chamada anterior MQCONN ou MQCONNX.

### **HOBJ (número inteiro assinado de 10 dígitos)-entrada**

Manipulação de objetos.

Essa manipulação representa o objeto da fila com atributos que devem ser configurados. A manipulação foi retornada por uma chamada MQOPEN anterior que especificou a opção OOSSET.

### **SELCNT (número inteiro assinado de 10 dígitos)-entrada**

Contagem de seletores

Essa é a contagem de seletores fornecidos na matriz SELS. É o número de atributos a serem configurados. Zero é um valor válido. O número máximo permitido é 256..

### **(Número inteiro assinado de 10 dígitos x SELCNT)-entrada**

Matriz de seletores de atributo..

Esta é uma matriz de seletores de atributo **SELCNT**; cada seletor identifica um atributo (número inteiro ou caractere) com um valor que deve ser configurado.

Cada seletor deve ser válido pelo tipo de fila que HOBJ representa. Apenas determinados valores de IA\* e CA\* são permitidos; esses valores são listados posteriormente nesta seção.

Os seletores podem ser especificados em qualquer ordem. Os valores de atributo que correspondem a seletores de atributo de número inteiro (seletores IA\*) devem ser especificados em INTATR na mesma ordem em que esses seletores ocorrem em SELS. Os valores de atributo que correspondem aos seletores de atributo de caractere (seletores CA\*) devem ser especificados em CHRATR na mesma ordem em que esses seletores ocorrem.. Seletores IA\* podem ser intercalados com os seletores CA\*; apenas a ordem relativa dentro de cada tipo é importante.

Não é um erro especificar o mesmo seletor mais de uma vez; se isso for feito, o último valor especificado para um seletor específico será aquele que entra em vigor..

#### **Nota:**

1. Os seletores de atributo de número inteiro e de caractere são alocados dentro de dois intervalos diferentes; os seletores IA\* residem no intervalo IAFRST até IALAST e os seletores CA\* dentro do intervalo CAFRST até CALAST.

Para cada intervalo, as constantes IALSTU e CALSTU definem o valor mais alto que o Gerenciador de Filas aceitará.

2. Se todos os seletores IA\* ocorrerem primeiro, os mesmos números de elementos poderão ser usados para endereçar os elementos correspondentes nas matrizes SELS e INTATR.

Os atributos que podem ser configurados são listados na tabela a seguir: Nenhum outro atributo pode ser configurado utilizando esta chamada. Para os seletores de atributo CA\*, a constante que define o comprimento em bytes da cadeia necessária em CHRATR é fornecida entre parênteses.

<i>Tabela 751. seletores de atributos MQSET para filas</i>		
<b>Seletor</b>	<b>Descrição</b>	<b>Nota</b>
CATRGD	Dados do acionador (LNTRGD).	<u>“2” na página 1399</u>
IADIST	Suporte à lista de distribuições	<u>“1” na página 1399</u>
IAIGET	Se operações get são permitidas.	
IAIPUT	Se operações put são permitidas.	
IATRGC	Acionador de controle.	<u>“2” na página 1399</u>
IATRGD	Profundidade do acionador.	<u>“2” na página 1399</u>
IATRGP	Prioridade da mensagem limite para acionadores.	<u>“2” na página 1399</u>
IATRGT	Tipo de acionador.	<u>“2” na página 1399</u>

**Notas:**

1. Suportado apenas nas seguintes plataformas:

-  AIX
-  IBM i
-  Windows

e para clientes IBM MQ conectados a esses sistemas.

2. Não suportado em VSE/ESA..

**IACNT (número inteiro assinado de 10 dígitos)-entrada**

Contagem de atributos de número inteiro.

Esse é o número de elementos na matriz INTATR e deve ser pelo menos o número de seletores IA\* no parâmetro **SELS** . Zero é um valor válido se não houver nenhum.

**INTATR (número inteiro assinado de 10 dígitos x rxIACNT)-entrada**

Matriz de atributos de número inteiro

Essa é uma matriz de valores de atributo de número inteiro IACNT . Esses valores de atributos devem estar na mesma ordem que os seletores IA\* na matriz SELS .

**CALEN (número inteiro assinado de 10 dígitos)-entrada**

O comprimento do buffer de atributos de caracteres

Esse é o comprimento em bytes do parâmetro **CHRATR** e deve ser pelo menos a soma dos comprimentos dos atributos de caractere especificados na matriz SELS . Zero será um valor válido se não houver seletores CA\* em SELS.

### **CHRATR (cadeia de caracteres de 1 byte x CALEN)-entrada**

Atributos de caractere..

Este é o buffer que contém os valores de atributo de caractere, concatenados juntos O comprimento do buffer é fornecido pelo parâmetro **CALEN** .

Os atributos de caracteres devem ser especificados na mesma ordem dos seletores CA\* na matriz SELS . O comprimento de cada atributo de caractere é fixo (consulte SELS). Se o valor a ser configurado para um atributo contiver menos caracteres não em branco do que o comprimento definido do atributo, o valor em CHRATR deverá ser preenchido à direita com espaços em branco para que o valor do atributo corresponda ao comprimento definido do atributo.

### **CMPCOD (número inteiro assinado de 10 dígitos)-saída**

Código de conclusão.

É um dos seguintes:

#### **CCOK**

Indica conclusão bem-sucedida.

#### **CCFAIL**

A chamada falhou.

### **REASON (número inteiro assinado de 10 dígitos)-saída**

Código de razão qualificando CMPCOD.

Se CMPCOD for CCOK:

#### **RCNONE**

(0, X'000') Nenhuma razão a relatar.

Se CMPCOD for CCFAIL:

#### **RC2219**

(2219, X'8AB') chamada MQI reinserida antes da chamada anterior ser concluída.

#### **RC2006**

(2006, X'7D6') Comprimento de atributos de caractere não válido.

#### **RC2007**

(2007, X'7D7') Sequência de atributos de caractere não válida.

#### **RC2009**

(2009, X'7D9') Conexão ao gerenciador de filas perdida.

#### **RC2018**

(2018, X'7E2') Manipulação de conexões não válida.

#### **RC2019**

(2019, X'7E3') Manipulação de objetos não válida.

#### **RC2020**

(2020, X'7E4') O valor para o atributo da fila inibir-obter ou inibir-colocar não é válido.

#### **RC2021**

(2021, X'7E5') Contagem de atributos de número inteiro inválida.

#### **RC2023**

(2023, X'7E7') A matriz de atributos de número inteiro não é válida.

#### **RC2040**

(2040, X'7F8') Fila não aberta para configurar.

#### **RC2041**

(2041, X'7F9') Definição de objeto alterada desde a abertura.

**RC2101**

(2101, X'835') Objeto danificado.

**RC2052**

(2052, X'804') A fila foi excluída.

**RC2058**

(2058, X'80A') Nome do gerenciador de filas não válido ou não conhecido.

**RC2059**

(2059, X'80B') Gerenciador de filas não disponível para conexão.

**RC2162**

(2162, X'872') Gerenciador de filas sendo encerrado.

**RC2102**

(2102, X'836') Recursos insuficientes do sistema disponíveis.

**RC2065**

(2065, X'811 ') Contagem de seletores inválida.

**RC2067**

(2067, X'813 ') Seletor de atributo inválido.

**RC2066**

(2066, X'812 ') Contagem de seletores muito grande.

**RC2071**

(2071, X'817') Armazenamento insuficiente disponível.

**RC2075**

(2075, X'81B') O valor para o atributo trigger-control não é válido.

**RC2076**

(2076, X'81C') Valor para atributo de profundidade do acionador não válido.

**RC2077**

(2077, X'81D') Valor para o atributo trigger-message-priority não válido.

**RC2078**

(2078, X'81E') O valor para o atributo trigger-type não é válido.

**RC2195**

(2195, X'893') Ocorreu um erro inesperado.

**Declaração de RPG**

```

C*..1.....2.....3.....4.....5.....6.....7..
C          CALLP      MQSET(HCONN : HOBJ : SELCNT :
C                      SELS(1) : IACNT : INTATR(1) :
C                      CALEN : CHRATR : CMPCOD :
C                      REASON)

```

A definição de protótipo para a chamada é:

```

D*..1.....2.....3.....4.....5.....6.....7..
DMQSET      PR          EXTPROC('MQSET')
D* Connection handle
D HCONN          10I 0 VALUE
D* Object handle
D HOBJ          10I 0 VALUE
D* Count of selectors
D SELCNT        10I 0 VALUE
D* Array of attribute selectors
D SELS          10I 0
D* Count of integer attributes
D IACNT         10I 0 VALUE
D* Array of integer attributes
D INTATR        10I 0
D* Length of character attributes buffer
D CALEN         10I 0 VALUE
D* Character attributes

```

D CHRATR	*	VALUE
D* Completion code		
D CMPCOD	10I	0
D* Reason code qualifying CMPCOD		
D REASON	10I	0

## IBM i MQSETMP (Configurar propriedade do identificador de mensagem) em IBM i .

A chamada MQSETMP configura ou modifica uma propriedade de um identificador de mensagem

- [“Sintaxe” na página 1402](#)
- [“Observações de Uso” na página 1402](#)
- [“Parâmetros” na página 1404](#)
- [“Declaração de RPG” na página 1406](#)

### Sintaxe

MQSETMP (*Hconn, Hmsg, SetPropOpts, Name, PropDesc, Type, ValueLength, Value, CompCode, Reason*)

### Observações de Uso

- É possível usar essa chamada apenas quando o próprio gerenciador de filas coordena a unidade de trabalho. Ele pode ser:
  - Uma unidade de trabalho local, em que as mudanças afetam apenas os recursos do IBM MQ
  - Uma unidade de trabalho global, na qual as mudanças podem afetar recursos pertencentes a outros gerenciadores de recursos, bem como afetar recursos do IBM MQ .

Para obter detalhes adicionais sobre as unidades de trabalho locais e globais, consulte [“MQBEGIN \(Iniciar unidade de trabalho\) em IBM i” na página 1299](#)
- Em ambientes nos quais o gerenciador de filas não coordena a unidade de trabalho, use a chamada de retorno apropriada, em vez de MQBACK. O ambiente também pode suportar um retorno implícito causado pela finalização anormal do aplicativo.
  - No z/OS, use as seguintes chamadas:
    - Programas em lote (incluindo IMS programas DL/I em lote) podem usar a chamada MQBACK se a unidade de trabalho afetar apenas recursos IBM MQ . No entanto, se a unidade de trabalho afetar recursos do IBM MQ e recursos pertencentes a outros gerenciadores de recursos (por exemplo, Db2 ), use a chamada SRRBACK fornecida pelo z/OS Recoverable Resource Service (RRS). A chamada SRRBACK retorna mudanças em recursos pertencentes aos gerenciadores de recursos que foram ativados para coordenação RRS.
    - Aplicativos CICS devem usar o comando EXEC CICS SYNCPOINT ROLLBACK para voltar a unidade de trabalho. Não use a chamada MQBACK para os aplicativos CICS
    - Os aplicativos IMS (que não sejam programas DL/I em lote) devem usar chamadas IMS como ROLB para voltar a unidade de trabalho. Não use a chamada MQBACK para aplicativos IMS (exceto programas DL/I em lote).
  - No IBM i, use essa chamada para unidades locais de trabalho coordenadas pelo gerenciador de fila.. Isso significa que uma definição de confirmação não deve existir no nível da tarefa, ou seja, o comando STRCMTCTL com o parâmetro **CMTSCOPE (\*JOB)** não deve ter sido emitido para a tarefa
- Se um aplicativo terminar com mudanças não confirmados em uma unidade de trabalho, a disposição dessas mudanças dependerá se o aplicativo será encerrado normalmente ou de forma anormal. Consulte as notas de uso em [“MQDISC \(Desconectar o gerenciador de filas\) no IBM i” na página 1336](#) para obter detalhes adicionais

- Quando um aplicativo coloca ou obtém mensagens em grupos ou segmentos de mensagens lógicas, o gerenciador de filas retém informações relacionadas ao grupo de mensagens e mensagens lógicas para as últimas chamadas MQPUT e MQGET bem-sucedidas. Essas informações estão associadas ao identificador de filas e incluem itens como:
  - Os valores dos campos *GroupId*, *MsgSeqNumber*, *Offset* e *MsgFlags* em MQMD..
  - Indica se a mensagem faz parte de uma unidade de trabalho
  - Para a chamada MQPUT: se a mensagem é persistente ou não persistente.

O gerenciador de filas mantém três conjuntos de informações de grupo e segmento, um conjunto para cada um dos seguintes:

- A última chamada MQPUT bem-sucedida (isso pode fazer parte de uma unidade de trabalho)
- A última chamada MQGET bem-sucedida que removeu uma mensagem da fila (isso pode fazer parte de uma unidade de trabalho).
- A última chamada MQGET bem-sucedida que navegou em uma mensagem na fila (isso não pode fazer parte de uma unidade de trabalho)

Se o aplicativo colocar ou receber as mensagens como parte de uma unidade de trabalho e o aplicativo decidir restaurar a unidade de trabalho, as informações do grupo e do segmento serão restauradas para o valor que ele tinha anteriormente:

- As informações associadas à chamada MQPUT são restauradas para o valor que ela tinha antes da primeira chamada MQPUT bem-sucedida para essa manipulação de fila na unidade de trabalho atual..
- As informações associadas à chamada MQGET são restauradas para o valor que ela tinha antes da primeira chamada MQGET bem-sucedida para essa manipulação de fila na unidade de trabalho atual...

As filas que foram atualizadas pelo aplicativo após a unidade de trabalho ter iniciado, mas fora do escopo da unidade de trabalho, não terão suas informações de grupo e de segmento restauradas se a unidade de trabalho for restaurada.

A restauração das informações do grupo e do segmento para seu valor anterior quando uma unidade de trabalho é restaurada permite que o aplicativo difunda um grupo de mensagens grandes ou uma mensagem lógica grande que consiste em muitos segmentos em várias unidades de trabalho e reinicie no ponto correto no grupo de mensagens ou mensagem lógica se uma das unidades de trabalho falhar.

Usar várias unidades de trabalho pode ser vantajoso se o gerenciador de fila local tiver apenas armazenamento de fila limitado. No entanto, o aplicativo deve manter informações suficientes para poder reiniciar a colocação ou obtenção de mensagens no ponto correto se ocorrer uma falha do sistema.

Para obter detalhes de como reiniciar no ponto correto após uma falha do sistema, consulte a opção PMLOGO descrita em PMOPT (número inteiro assinado de 10 dígitos) e a opção GMLOGO descrita em GMOPT (número inteiro assinado de 10 dígitos).

As notas de uso restantes se aplicam apenas quando o gerenciador de filas coordena as unidades de trabalho:

- Uma unidade de trabalho tem o mesmo escopo que uma manipulação de conexões.. Todas as chamadas de IBM MQ que afetam uma unidade de trabalho específica devem ser executadas usando a mesma manipulação de conexões Chamadas emitidas usando um identificador de conexão diferente (por exemplo, chamadas emitidas por outro aplicativo) afetam uma unidade de trabalho diferente. Consulte HCONN (número inteiro assinado de 10 dígitos)-saída para obter informações sobre o escopo de identificadores de conexões...
- Apenas as mensagens que foram colocadas ou recuperadas como parte da unidade de trabalho atual são afetada por esta chamada
- Um aplicativo de longa execução que emite chamadas MQGET ou MQPUT ou MQPUT1 dentro de uma unidade de trabalho, mas que nunca emite uma confirmação ou chamada de restauração, pode preencher filas com mensagens que não estão disponíveis para outros aplicativos. Para

proteger contra essa possibilidade, o administrador deve configurar o atributo do gerenciador de filas **MaxUncommittedMsgs** para um valor que seja baixo o suficiente para evitar que os aplicativos runaway preencham as filas, mas alto o suficiente para permitir que os aplicativos do sistema de mensagens esperados funcionem corretamente.

## Parâmetros

A chamada MQSETMP possui os seguintes parâmetros:

### HCONN (número inteiro assinado de 10 dígitos)-entrada

Esta manipulação representa a conexão ao gerenciador de filas.

O valor deve corresponder ao identificador de conexão que foi usado para criar o identificador de mensagem especificado no parâmetro **HMSG** ...

Se o identificador de mensagem foi criado usando HCUNAS, uma conexão válida deverá ser estabelecida na configuração de encadeamento de uma propriedade do identificador de mensagem, caso contrário, a chamada falhará com o código de razão RC2009

### HMSG (número inteiro assinado de 20 dígitos)-entrada

Esta é a manipulação de mensagem a ser modificada O valor foi retornado por uma chamada MQCRTMH anterior.

### SETOPT (MQSMPO)-entrada

Controle como as propriedades de mensagem são configuradas

Essa estrutura permite que os aplicativos especifiquem opções que controlam como as propriedades de mensagem são configuradas A estrutura é um parâmetro de entrada na chamada MQSETMP.. Consulte [MQSMPO](#) para obter informações adicionais

### PRNAME (MQCHARV)-entrada

Este é o nome da propriedade a ser configurada

Consulte [Nomes de Propriedade](#) e [Restrições de Nome de Propriedade](#) para obter informações adicionais sobre o uso de nomes de propriedade

### PRPDSC (MQPD)-entrada/saída

Esta estrutura é utilizada para definir os atributos de uma propriedade, incluindo:

- o que acontece se a propriedade não for suportada
- a qual contexto de mensagem a propriedade pertence
- para quais mensagens a propriedade é copiada à medida que flui

Consulte [MQPD](#) para obter informações adicionais sobre esta estrutura..

### TYPE (número inteiro assinado de 10 dígitos)-entrada

O tipo de dados da propriedade sendo configurada. Pode ser um dos seguintes:

#### TYPBOL

Um booleano. *ValueLength* deve ser 4.

#### TYPBST

Uma sequência de bytes. *ValueLength* deve ser zero ou maior.

#### TYPI8

Um número inteiro assinado de 8 bits. *ValueLength* deve ser 1.

#### TYPI16

Um número inteiro assinado de 16 bits. *ValueLength* deve ser 2.

#### TYPI32

Um número inteiro assinado de 32 bits. *ValueLength* deve ser 4.



**TYP164**

Um número inteiro assinado de 64 bits. *ValueLength* deve ser 8..

**TYPF32**

Um número de vírgula flutuante de 32 bits.. *ValueLength* deve ser 4.

**TYPF64**

Um número de vírgula flutuante de 64 bits.. *ValueLength* deve ser 8..

**TYPSTR**

Uma sequência de Caracteres *ValueLength* deve ser zero ou maior ou o valor especial VLNULL.

**TYPNUL**

A propriedade existe mas possui um valor nulo. *ValueLength* deve ser zero.

**VALLEN (número inteiro assinado de 10 dígitos)-entrada**

O comprimento em bytes do valor da propriedade no parâmetro *Value* .

Zero é válido apenas para valores nulos ou para sequências ou sequências de bytes. Zero indica que a propriedade existe, mas que o valor não contém caracteres ou bytes.

O valor deve ser maior ou igual a zero ou o valor especial a seguir se o parâmetro *Type* tiver TYPSTR configurado:

**VLNULL**

O valor é delimitado pelo primeiro nulo encontrado na cadeia. O nulo não é incluído como parte da cadeia. Esse valor será inválido se TYPSTR também não estiver configurado.

Nota: O caractere nulo usado para finalizar uma sequência se VLNULL for configurado é um nulo do conjunto de caracteres do Valor.

**VALUE (cadeia de bits de 1 byte x VALLEN)-entrada**

O valor da propriedade a ser configurada O buffer deve ser alinhado em um limite apropriado para a natureza dos dados no valor..

Na linguagem de programação C, o parâmetro é declarado como um ponteiro para vazio; o endereço de qualquer tipo de dados pode ser especificado como o parâmetro.

Se *ValueLength* for zero, *Value* não será referido. Nesse caso, o endereço de parâmetro transmitido por programas gravados em C ou assembler System/390 pode ser nulo.

**CMPCOD (número inteiro assinado de 10 dígitos)-saída**

O código de conclusão; é um dos seguintes:

**CCOK**

Indica conclusão bem-sucedida.

**CCFAIL**

A chamada falhou.

**REASON (número inteiro assinado de 10 dígitos)-saída**

O código de razão que qualifica *CMPCOD*.

Se *CMPCOD* for CCOK:

**RCNONE**

(0, X'000') Nenhuma razão a relatar.

Se *CMPCOD* for CCWARN:

**RC2421**

(2421, X'0975 ') Uma pasta MQRFH2 contendo propriedades não pôde ser analisada.

Se *CMPCOD* for CCFAIL:

**RC2204**

(2204, X'089C') Adaptador não disponível.

**RC2130**

(2130, X'852') Não foi possível carregar o módulo de serviço do adaptador.

**RC2157**

(2157, X'86D') ASIDs de início e primárias são diferentes.

**RC2004**

(2004, X'07D4') Parâmetro de valor inválido.

**RC2005**

(2005, X'07D5') Parâmetro de comprimento do valor inválido.

**RC2219**

(2219, X'08AB') chamada MQI inserida antes da chamada anterior ser concluída.

**RC2460**

(2460, X'099C') Ponteiro do identificador de mensagens inválido.

**RC2499**

(2499, X'09C3') Identificador de mensagem já em uso.

**RC2046**

(2046, X'07FE') Opções não válidas ou não consistentes.

**RC2482**

(2482, X'09B2') Estrutura do descritor de propriedade inválida.

**RC2442**

(2442, X'098A') Nome da propriedade inválido

**RC2473**

(2473, X'09A9') Tipo de dados de propriedade inválido.

**RC2472**

(2472, X'09A8') Erro de formato numérico encontrado nos dados de valor...

**RC2463**

(2463, X'099F') A estrutura das opções da propriedade de mensagens não é válida...

**RC2111**

(2111, X'083F') Identificador do conjunto de caracteres codificados do nome da propriedade inválido.

**RC2071**

(2071, X'817') Armazenamento insuficiente disponível.

**RC2195**

(2195, X'893') Ocorreu um erro inesperado.

Consulte [“Códigos de retorno para IBM i \(ILE RPG\)”](#) na página 1477 para obter mais detalhes.

## Declaração de RPG

```

C*.1.....2.....3.....4.....5.....6.....7..
C          CALLP      MQSETMP(HCONN : HMSG : SETOPT :
                          PRNAME : PRPDSC :
                          TYPE : VALLEN : VALUE :
                          CMPCOD : REASON)

```

A definição de protótipo para a chamada é:

```

DMQSETMP      PR          EXTPROC('MQSETMP')
D* Connection handle
D HCONN              10I 0 VALUE
D* Message handle
D HMSG              10I 0 VALUE
D* Options that control the action of MQSETMP
D SETOPT            20A
D* Property name
D PRNAME            32A
D* Property descriptor

```

D PRPDSC	24A
D* Property data type	
D TYPE	10I 0 VALUE
D* Length of the Value area	
D VALLEN	10I 0 VALUE
D* Property value	
D VALUE	* VALUE
D* Completion code	
D CMPCOD	10I 0
D* Reason code qualifying CompCode	
D REASON	10I 0

## IBM i MQSTAT (recuperar informações de status) em IBM i

Use a chamada MQSTAT para recuperar as informações de status. O tipo de informações de status retornado é determinado pelo valor STYPE especificado na chamada.

- [“Sintaxe” na página 1407](#)
- [“Observações de Uso” na página 1407](#)
- [“Parâmetros” na página 1407](#)
- [“Declaração de RPG” na página 1408](#)

### Sintaxe

MQSTAT (*HCONN*, *STYPE*, *STAT*, *CMPCOD*, *REASON*)

### Observações de Uso

1. Uma chamada para MQSTAT especificando um tipo de STATAPT retorna informações sobre operações MQPUT e MQPUT1 assíncronas anteriores. A estrutura MQSTAT transmitida na chamada é concluída com as primeiras informações de aviso assíncrono ou de erro registradas para essa conexão. Se erros ou avisos adicionais seguirem o primeiro, eles normalmente não alterarão esses valores. No entanto, se ocorrer um erro com um código de conclusão CCWARN, uma falha subsequente com um código de conclusão CCFAIL será retornada em vez disso.
2. Se nenhum erro tiver ocorrido desde que a conexão foi estabelecida ou desde a última chamada para MQSTAT, um CMPCOD de CCOK e REASON de RCNONE serão retornados.
3. As contagens do número de chamadas assíncronas que foram processadas no identificador de conexão são retornadas usando três contadores; STSPSC, STSPWC e STSPFC. Esses contadores são incrementados pelo gerenciador de filas toda vez que uma operação assíncrona é processada com êxito, possui um aviso ou falha (observe que, para fins de contabilidade, uma colocação em uma lista de distribuição conta uma vez por fila de destino em vez de uma vez por lista de distribuição).
4. Uma chamada bem-sucedida para MQSTAT resulta em quaisquer informações de erro ou contagens anteriores sendo reconfiguradas.

### Parâmetros

A chamada MQSTAT possui os seguintes parâmetros:

#### Hconn (MQHCONN)-entrada

Esta manipulação representa a conexão ao gerenciador de filas. O valor de *Hconn* foi retornado por uma chamada anterior MQCONN ou MQCONNX.

#### STYPE (número inteiro assinado de 10 dígitos)-entrada

Tipo de informações de status sendo solicitadas. O único valor válido é:

##### STATAPT

Retornar informações sobre operações de colocação assíncronas anteriores

## **STS (MQSTS)-entrada/saída**

Estrutura de informações de status Consulte [“MQSTS \(Estrutura de Relatório de Status\) no IBM i”](#) na página 1275 para obter detalhes.

## **CMPCOD (número inteiro assinado de 10 dígitos)-saída**

O código de conclusão; é um dos seguintes:

### **CCOK**

Indica conclusão bem-sucedida.

### **CCFAIL**

A chamada falhou.

## **REASON (número inteiro assinado de 10 dígitos)-saída**

O código de razão qualificando *CMPCOD*.

Se *CMPCOD* for CCOK:

### **RCNONE**

(0, X'000') Nenhuma razão a relatar.

Se *CMPCOD* for CCFAIL:

### **RC2374**

(2374, X' 946 ') A saída da API falhou

### **RC2183**

(2183, X'887') Não foi possível carregar a saída de API.

### **RC2219**

(2219, X'8AB') Chamada MQI inserida antes da chamada anterior concluída.

### **RC2009**

(2009, X'7D9') Conexão ao gerenciador de filas perdida.

### **RC2203**

(2203, X'89B') Conexão sendo encerrada.

### **RC2018**

(2018, X'7E2') Manipulação de conexões não válida.

### **RC2162**

(2162, X'872 ') Gerenciador de filas parando

### **RC2102**

(2102, X'836') Recursos insuficientes do sistema disponíveis.

### **RC2430**

(2430, X'97E') Erro com o tipo MQSTAT.

### **RC2071**

(2071, X'817') Armazenamento insuficiente disponível.

### **RC2424**

(2424, X' 978 ') Erro com a estrutura MQSTS

### **RC2195**

(2195, X'893') Ocorreu um erro inesperado.

### **RC2298**

(2298, X'8FA') A função solicitada não está disponível no ambiente atual.

Para obter informações detalhadas sobre esses códigos, consulte:

- [Mensagens e códigos de razão](#)

## **Declaração de RPG**

C\*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7

```

C          CALLP      MQSTAT(HCONN : ETYPE : ERR :
C          CMPCOD : REASON)

```

A definição de protótipo para a chamada é:

```

D.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7
DMQSTAT      PR          EXTPROC('MQSTAT')
D* Connection handle
D HCONN          10I 0 VALUE
D* Status information type
D STYPE          10I 0 VALUE
D* Status information
D STATUS        296A
D* Completion code
D CMPCOD        10I 0
D* Reason code qualifying CompCode
D REASON        10I 0

```

## IBM i MQSUB (Assinatura de Registro) em IBM i

A chamada MQSUB registra a assinatura de aplicativos para um tópico específico

- [“Sintaxe” na página 1409](#)
- [“Observações de Uso” na página 1409](#)
- [“Parâmetros” na página 1411](#)
- [“Declaração de RPG” na página 1414](#)

### Sintaxe

MQSUB (*HCONN*, *SUBDSC*, *HOBJ*, *HSUB*, *CMPCOD*, *REASON*)

### Observações de Uso

- A assinatura é feita para um tópico, nomeado usando o nome abreviado de um objeto do tópico predefinido, o nome completo da sequência de tópicos ou é formado pela concatenação de duas partes, conforme descrito em [Combinando sequências de tópicos](#).
- O gerenciador de filas executa verificações de segurança quando uma chamada MQSUB é emitida, para verificar se o identificador do usuário sob o qual o aplicativo é executado possui o nível apropriado de autoridade antes que o acesso seja permitido. O objeto do tópico apropriado está localizado por um nome abreviado sendo fornecido na chamada ou o objeto do nome abreviado mais próximo na hierarquia de tópicos sendo localizado se um nome longo for fornecido. Uma verificação de autoridade é feita neste objeto do tópico para assegurar que a autoridade para assinar seja configurada e na fila de destino para assegurar que a autoridade para a saída seja configurada. Se a opção SDMAN for usada, isso significa que uma verificação de autoridade será feita no nome da fila gerenciada associada a esse objeto de tópico e, se uma fila não gerenciada for fornecida, isso significa que uma verificação de autoridade será feita na fila representada pelo parâmetro **HOBJ**
- O *HOBJ* retornado na chamada MQSUB quando a opção SOMAN é usada, pode ser consultado para descobrir atributos como o limite de restauração e o nome de reenfileiramento de restauração excessiva. Também é possível consultar o nome da fila gerenciada, mas você não deve tentar abrir diretamente essa fila
- As assinaturas podem ser agrupadas permitindo que apenas uma única publicação seja entregue ao grupo de assinaturas mesmo quando mais de um grupo tiver correspondido à publicação. As assinaturas são agrupadas usando a opção SOGRP e, para agrupar assinaturas, elas devem:
  - esteja usando a mesma fila nomeada (que não está usando a opção SOMAN) no mesmo gerenciador de filas-representado pelo parâmetro **HOBJ** na chamada MQSUB
  - compartilhar o mesmo *SDCID*
  - ser do mesmo *SDSL*

Esses atributos definem o conjunto de assinaturas consideradas no grupo e também são os atributos que não podem ser alterados, se uma assinatura for agrupada. A alteração de *SDSL* resulta em RC2512e a alteração de qualquer um dos outros (que pode ser alterado se uma assinatura não for agrupada) resulta em RC2515.

- Os campos no MQSD são preenchidos no retorno de uma chamada MQSUB que usa a opção SORES. O MQSD retornado pode ser transmitido diretamente para uma chamada MQSUB que usa a opção SOALT com quaisquer mudanças que você precisa fazer na assinatura aplicada ao MQSD.. Alguns campos possuem considerações especiais conforme observado na tabela.

Tabela 752. Saída MQSD do MQSUB	
Nome do campo em MQSD	Considerações Especiais
Opções de Acesso ou Criação	Nenhuma dessas opções é configurada no retorno da chamada MQSUB.. Se, posteriormente, você reutilizar o MQSD em uma chamada do MQSUB, a opção necessária deverá ser configurada explicitamente.
Opções de Durabilidade, opções de Destino, opções de Registro & opções de Curinga	Estas opções serão configuradas conforme apropriado
Opções de publicação	Essas opções serão configuradas conforme apropriado, exceto para SONEWP que é aplicável apenas ao SOCRE.
Outras Opções	Essas opções ficam inalteradas no retorno de uma chamada MQSUB. Elas controlam como a chamada de API é emitida e não são armazenadas com a assinatura. Elas devem ser configuradas conforme necessário em qualquer chamada MQSUB subsequente reutilizando o MQSD.
ObjectName	Este campo de entrada apenas fica inalterado no retorno de uma chamada MQSUB.
ObjectString	Este campo de entrada apenas fica inalterado no retorno de uma chamada MQSUB. O nome do tópico completo usado será retornado no campo <i>SDRO</i> , se um buffer for fornecido
AlternateUserId e AlternateSecurityId	Esses campos de entrada apenas ficam inalterados no retorno de uma chamada MQSUB. Elas controlam como a chamada de API é emitida e não são armazenadas com a assinatura. Elas devem ser configuradas conforme necessário em qualquer chamada MQSUB subsequente reutilizando o MQSD.
SubExpiry	No retorno de uma chamada MQSUB usando a opção SORES, este campo será configurado para a expiração original da assinatura e não para o tempo de expiração restante Se você então reutilizar o MQSD em uma chamada MQSUB usando a opção SOALT, reconfigurará a expiração da assinatura para iniciar a contagem regressiva novamente.
SubName	Este campo é um campo de entrada em uma chamada MQSUB e não é alterado na saída.
SubUserData e SelectionString	Esses campos de comprimento variável serão retornados na saída de uma chamada MQSUB usando a opção SORES, se um buffer for fornecido e também um comprimento de buffer positivo em <i>VCHRP</i> . Se nenhum buffer for fornecido, apenas o comprimento será retornado no campo <i>VCHRL</i> do MQCHARV.If o buffer fornecido for menor do que o espaço necessário para retornar o campo, apenas <i>VCHRP</i> bytes serão retornadas no buffer fornecido  Se posteriormente você reutilizar o MQSD em uma chamada MQSUB usando a opção SOALT e um buffer não for fornecido, mas um <i>VCHRL</i> diferente de zero for fornecido, se esse comprimento corresponder ao comprimento existente do campo, nenhuma alteração será feita no campo.

Tabela 752. Saída MQSD do MQSUB (continuação)	
Nome do campo em MQSD	Considerações Especiais
SubCorrelId e PubAccountingToken	Se você não usar SOSCID, o <i>SDCID</i> será gerado pelo gerenciador de filas. Se você não usar SOSETI, o <i>SDACC</i> será gerado pelo gerenciador de filas.  Esses campos serão retornados no MQSD de uma chamada MQSUB usando a opção SORES. Se eles forem gerados pelo gerenciador de filas, o valor gerado será retornado em uma chamada MQSUB usando a opção SOCRE ou SOALT.
PubPriority, SubLevel & PubApplIdentityData	Esses campos serão retornados no MQSD
ResObjectString	Esse campo somente de saída será retornado no MQSD, se um buffer for fornecido

## Parâmetros

A chamada MQSUB possui os parâmetros a seguir:

### HCONN (número inteiro assinado de 10 dígitos)-entrada

Esta manipulação representa a conexão ao gerenciador de filas. O valor de *HCONN* foi retornado por uma chamada anterior MQCONN ou MQCONNX.

### SUBDSC (MQSD)-entrada/saída

Essa é uma estrutura que identifica o objeto com uso sendo registrado pelo aplicativo. Consulte “MQSD (Descritor de assinatura) em IBM i” na página 1257 para obter mais informações..

### HOBJ (número inteiro assinado de 10 dígitos)-entrada/saída

Esta manipulação representa o acesso que foi estabelecido para obter as mensagens enviadas a essa assinatura. Essas mensagens podem ser armazenadas em uma fila específica ou o gerenciador de fila pode ser solicitado a gerenciar seu armazenamento sem a necessidade de uma fila específica..

Manipulação de objetos.

Se uma fila específica for usada, ela deverá ser associada à assinatura no momento da criação. Isso pode ser feito de duas maneiras:

- Ao fornecer essa manipulação ao chamar MQSUB com a opção SDCRT, Se esse identificador for fornecido como um parâmetro de entrada na chamada, ele deverá ser um identificador de objetos válido retornado de uma chamada MQOPEN anterior de uma fila usando pelo menos uma das opções OOINP\*, OOOUT (se uma fila remota, por exemplo) ou OOBRW.. Se esse não for o caso, a chamada falha com RC2019. Ele não pode ser uma manipulação de objetos para uma fila de alias que é resolvida para um objeto de tópico Se sim, a chamada falhará com RC2019
- Usando o comando DEFINE SUB MQSC e fornecendo a esse comando o nome de um objeto de fila.

Se o gerenciador de filas for gerenciar o armazenamento de mensagens enviadas para esta assinatura, será necessário indicar isso quando a assinatura for criada, usando a opção SOMAN e configurando o valor de parâmetro como HONONE.. O gerenciador de filas retorna o identificador como um parâmetro de saída na chamada e o identificador que é retornado é conhecido como um identificador gerenciado. Se HONONE for especificado e SOMAN não for especificado, a chamada falhará com RC2019.

Um identificador gerenciado que é retornado pelo gerenciador de filas pode ser usado em uma chamada MQGET ou MQCB, com ou sem opções de procura, em uma chamada MQINQ ou em MQCLOSE. Ele não pode ser usado em MQPUT, MQSET ou em um MQSUB subsequente; a tentativa de fazer isso falha com RC2039 para MQPUT, RC2040 para MQSET ou RC2038 para MQSUB.

Se a opção SORES no campo *OPTS* na estrutura MQSD for usada para continuar essa assinatura, o identificador poderá ser retornado ao aplicativo nesse parâmetro se HONONE for especificado. É possível usar isso se a assinatura estiver usando uma manipulação gerenciada ou não Pode ser útil para assinaturas criadas usando DEFINE SUB se você desejar o identificador para a fila de assinaturas

definida no comando DEFINE SUB. No caso em que uma assinatura criada administrativamente está sendo continuada, a fila é aberta com OOINPQ e OOBW. Se outras opções forem necessárias, o aplicativo deverá abrir a fila de assinaturas explicitamente e fornecer a manipulação de objetos na chamada Se houver um problema ao abrir a fila, a chamada falhará com RC2522.. Se *HOBJ* for fornecido, ele deverá ser equivalente ao *HOBJ* na chamada MQSUB original. Isso significa que se uma manipulação de objetos retornada de uma chamada MQOPEN estiver sendo fornecida, a manipulação deverá estar na mesma fila usada anteriormente ou a chamada falhará com RC2019.

Se essa assinatura estiver sendo alterada, usando a opção SOALT no campo *OPTS* na estrutura MQSD, um *HOBJ* diferente poderá ser fornecido. Quaisquer publicações que foram entregues para a fila identificada anteriormente por meio desse parâmetro permanecem nessa fila e é responsabilidade do aplicativo recuperar essas mensagens se o parâmetro **HOBJ** agora representar uma fila diferente.

O uso desse parâmetro com várias opções de assinatura é resumido na tabela a seguir:

*Tabela 753. Usando o hobj com várias opções de assinatura*

Opções	Hobj	Descrição
SOCRT + SOMAN	Ignorado na entrada	Cria uma assinatura com o armazenamento de mensagens gerenciado pelo gerenciador de filas
SOCRT	Manipulação de objetos válida	Cria uma assinatura que fornece uma fila específica como destino para as mensagens.
SOES	HONONE	Continua uma assinatura criada anteriormente (gerenciada ou não) e faz o gerenciador de filas retornar a manipulação de objetos para uso pelo aplicativo.
SOES	Identificador de objeto válido, correspondente	Continua uma assinatura criada anteriormente que usa uma fila específica como o destino para mensagens e usa uma manipulação de objetos com opções de abertura específicas
SOALT + SOMAN	HONONE	Altera uma assinatura existente que estava anteriormente usando uma fila específica, para agora ser gerenciada
SOALT	Manipulação de objetos válida	Altera uma assinatura existente para usar uma fila específica (de gerenciada ou de uma fila específica diferente).

Se foi fornecido ou retornado, *HOBJ* deve ser especificado em chamadas MQGET subsequentes que você precisa para receber as publicações.

O identificador *HOBJ* cessa de ser válido quando a chamada MQCLOSE é emitida nele ou quando a unidade de processamento que define o escopo do identificador é finalizada O escopo de manipulação de objetos retornado é igual ao da manipulação de conexões especificada na chamada. Consulte HCONN para obter informações sobre o escopo de alça. Um MQCLOSE do identificador *HOBJ* não tem efeito no identificador *HSUB*.

### **HSUB (número inteiro assinado de 10 dígitos)-saída**

Esta manipulação representa a assinatura que foi feita. Ela pode ser usada para duas operações adicionais:

- Ele pode ser usado em uma chamada MQSUBRQ subsequente para solicitar que publicações sejam enviadas quando a opção SOPUBR tiver sido usada ao fazer a assinatura.
- Ela pode ser usada em uma chamada MQCLOSE subsequente para remover a assinatura que foi feita. A manipulação *HSUB* deixa de ser válida quando a chamada MQCLOSE é emitida ou quando a unidade de processamento que define o escopo da manipulação é encerrada. O escopo de manipulação de objetos retornado é igual ao da manipulação de conexões especificada na chamada. Um MQCLOSE do identificador *HSUB* não tem efeito no identificador *HOBJ*.



Esta manipulação não pode ser passada para uma chamada MQGET ou MQCB. Deve-se usar o parâmetro **HOB**J Transmitir essa manipulação para qualquer outra chamada IBM MQ resulta em RC2019.

### **CMPCOD (número inteiro assinado de 10 dígitos)-saída**

O código de conclusão; é um dos seguintes:

#### **CCOK**

Conclusão bem-sucedida

#### **CCWARN**

Aviso (conclusão parcial)

#### **CCFAIL**

Chamada com falha

### **REASON (número inteiro assinado de 10 dígitos)-saída**

O código de razão qualificando *CMPCOD*.

Se *CMPCOD* for CCOK:

#### **RCNONE**

(0, X'000') Nenhuma razão a relatar.

Se *CMPCOD* for CCFAIL:

#### **RC2019**

(2019 X'07E3') Identificador de objeto inválido

#### **RC2046**

(2046 X'07FE') Opções não válidas ou não consistentes

#### **RC2085**

(2085 X'0825 ') O objeto identificado não pode ser localizado.

#### **RC2161**

(2161 X'0871 ') quiesce do gerenciador de filas

#### **RC2298**

(2298 X'08FA') Função não suportada.

#### **RC2424**

(2424 X'0978 ') Descritor de assinatura (MQSD) não válido

#### **RC2425**

(2441 X' 979 ') Sequência de tópicos não válida

#### **RC2428**

(2428 X'097C') O nome da assinatura especificado não corresponde às assinaturas existentes

#### **RC2429**

(2429 X'097D') Nome da assinatura existe e está em uso por outro aplicativo

#### **RC2431**

(2431 X'097F') SubUserCampo de dados não válido

#### **RC2432**

(2432 X'0980 ') Existe uma assinatura

#### **RC2434**

(2434 X'0982 ') O nome da assinatura corresponde à assinatura existente

#### **RC2440**

(2440 X'0988 ') O campo SubName não é válido

#### **RC2441**

(2441 X'0989 ') O campo Objectstring não é válido

#### **RC2435**

(2435 X'0983 ') O atributo não pode ser alterado usando SDALT ou a assinatura foi criada com SDIMM.

**RC2436**

(2436 X'0984 ') A opção SODUR não é válida

**RC2459**

(2459, X'99B') Erro de sintaxe da sequência de seleção.

**RC2503**

(2503 X'09C7') chamadas MQSUB estão atualmente inibidas para os tópicos assinados.

**RC2519**

(2519, X'9D7') A sequência de seleção não é especificada na descrição de como usar uma estrutura MQCHARV.

**RC2551**

(2551, X'9F7') A sequência de seleção especificada não está disponível.

**Declaração de RPG**

```

C*..1.....2.....3.....4.....5.....6.....7..
C          CALLP      MQSUB(HCONN : SUBDSC : HOBJ :
C                               HSUB : CMPCOD : REASON)

```

A definição de protótipo para a chamada é:

```

D*..1.....2.....3.....4.....5.....6.....7..
DMQSUB      PR          EXTPROC('MQSUB')
D* Connection handle
D HCONN          10I 0 VALUE
D* Subscription descriptor
D SUBDSC          400A
D* Object handle for queue
D HOBJ          10I 0
D* Subscription object handle
D HSUB          10I 0
D* Completion code
D CMPCOD          10I 0
D* Reason code qualifying CompCode
D REASON          10I 0

```

**IBM i MQSUBRQ (Solicitação de Assinatura) no IBM i**

A chamada MQSUBRQ faz uma solicitação em uma assinatura..

- [“Sintaxe” na página 1414](#)
- [“Observações de Uso” na página 1414](#)
- [“Parâmetros” na página 1415](#)
- [“Declaração de RPG” na página 1416](#)

**Sintaxe**

MQSUBRQ (*HCONN*, *HSUB*, *ACTION*, *SUBROPT*, *CMPCOD*, *REASON*)

**Observações de Uso**

As notas de uso a seguir se aplicam ao uso de SRAPUB:

1. Se esse verbo for concluído com êxito, as publicações retidas correspondentes à assinatura especificada foram enviadas para a assinatura e podem ser recebidas usando MQGET ou MQCB usando o HOBJ retornado no verbo MQSUB original que criou a assinatura.
2. Se o tópico inscrito pelo verbo MQSUB original que criou a subscrição continha um curinga, mais de uma publicação retida pode ser enviada O número de publicações enviadas como resultado dessa chamada é registrado na estrutura SBROPT no campo *SRNMP* .

3. Se este verbo for concluído com um código de razão de RC2437 , não haverá publicações retidas atualmente para o tópico especificado..
4. Se esse verbo for concluído com um código de razão de RC2525 ou RC2526 , haverá publicações retidas atualmente para o tópico especificado, mas ocorreu um erro que significava que elas não puderam ser entregues.
5. O aplicativo deve ter uma assinatura atual para o tópico antes que ele possa fazer esta chamada. Se a assinatura foi feita em uma instância anterior do aplicativo e uma manipulação válida para a assinatura não está disponível, o aplicativo deve primeiro chamar MQSUB com a opção SORES para obter uma manipulação para ele para uso nessa chamada
6. As publicações são enviadas para o destino registrado para uso com a assinatura atual deste aplicativo. Se as publicações tiverem que ser enviadas para outro lugar, a subscrição deverá primeiro ser alterada usando a chamada MQSUB com a opção SOALT

## Parâmetros

A chamada MQSUBRQ possui os parâmetros a seguir:

### **HCONN (número inteiro assinado de 10 dígitos)-entrada**

Esta manipulação representa a conexão ao gerenciador de filas. O valor de *HCONN* foi retornado por uma chamada anterior MQCONN ou MQCONNX.

Em z/OS para aplicativos CICS , a chamada MQCONN pode ser omitida e o seguinte valor especificado para *HCONN*:

#### **HCDEFH**

Manipulação de conexões padrão.

### **HSUB (número inteiro assinado de 10 dígitos)-entrada**

Esse identificador representa a assinatura para a qual uma atualização deve ser solicitada O valor de *HSUB* foi retornado de uma chamada MQSUB anterior.

### **ACTION (número inteiro assinado de 10 dígitos)-entrada**

Esse parâmetro controla a ação específica que está sendo solicitada na assinatura.. Um (e apenas um) dos seguintes deve ser especificado:

#### **SRAPUB**

Esta ação solicita que uma publicação de atualização seja enviada para o tópico especificado.. Isso é normalmente usado se o assinante especificou a opção SOPUBR na chamada MQSUB quando ele fez a assinatura Se o gerenciador de filas tiver uma publicação retida para o tópico, isto será enviado ao assinante. Se não, a chamada falhará.. Se um aplicativo receber uma publicação que foi retida, isso será indicado pela propriedade da mensagem MQIsRetained dessa publicação.

Como o tópico na assinatura existente representada pelo parâmetro **HSUB** pode conter curingas, o assinante pode receber várias publicações retidas

### **SBROPT (MQSRO)-entrada/saída**

Essas opções controlam a ação de MQSUBRQ, consulte [“MQSRO-Opções de solicitação de assinatura”](#) na página 611 para obter detalhes..

### **CMPCOD (número inteiro assinado de 10 dígitos)-saída**

O código de conclusão; é um dos seguintes:

#### **CCOK**

Conclusão bem-sucedida

#### **CCWARN**

Aviso (conclusão parcial)

## CCFAIL

Chamada com falha

### Razão (número inteiro assinado de 10 dígitos)-saída

O código de razão qualificando *CMPCOD*.

Se *CPMPCOD* for CCOK:

## RCNONE

(0, X'000') Nenhuma razão a relatar.

Se *CPMPCOD* for CCFAIL:

## RC2298

2298 (X'08FA') A função solicitada não está disponível no ambiente atual.

## RC2437

2437 (X'0985 ') Não há publicações armazenadas atualmente para este tópico.

## RC2046

2046 (X'07FE') O parâmetro ou campo Options contém opções que não são válidas ou uma combinação das opções que não é válida.

## RC2161

2161 (X'0871 ') quiesce do gerenciador de filas

## RC2438

2438 (X'0986 ') Na chamada MQSUBRQ, o MQSRO Subscription Request Options não é válido.

## Declaração de RPG

```
C*..1.....2.....3.....4.....5.....6.....7..
C          CALLP      MQSUBRQ(HCONN : HSUB : ACTION :
C                               SBROPT : CMPCOD : REASON)
```

A definição de protótipo para a chamada é:

```
D*..1.....2.....3.....4.....5.....6.....7..
DMQSUBRQ      PR          EXTPROC('MQSUBRQ')
D* Connection handle
D HCONN              10I 0 VALUE
D* Subscription handle
D HSUB              10I 0 VALUE
D* Action requested on the subscription
D ACTION            10I 0 VALUE
D* Subscription Request Options
D SBROPT              16A
D* Completion code
D CMPCOD              10I 0
D* Reason code qualifying CompCode
D REASON            10I 0
```



## Atributos de objetos no IBM i

Esta coleção de tópicos lista apenas os objetos IBM MQ que podem ser o sujeito de uma chamada de função MQINQ e fornece detalhes dos atributos que podem ser consultados e os seletores a serem usados...

### Atributos para filas

Use estas informações para aprender sobre os diferentes tipos de definições de filas e os atributos suportados por cada um..

**Tipos de fila:** O gerenciador de filas suporta os seguintes tipos de definição de fila:

#### Fila local

Esta é uma fila física que armazena mensagens.. A fila existe no gerenciador de fila local.

Aplicativos conectados ao gerenciador de filas locais podem colocar mensagens e remover mensagens de filas desse tipo. O valor do atributo da fila **QType** é QTLOC.

### **Fila compartilhada**

Esta é uma fila física que armazena mensagens.. A fila existe em um repositório compartilhado acessível a todos os gerenciadores de filas que pertencem ao grupo de filas compartilhadas que possui o repositório compartilhado.

Aplicativos conectados a qualquer gerenciador de filas no grupo de filas compartilhadas podem colocar mensagens e remover mensagens de filas desse tipo. Essas filas são efetivamente iguais às filas locais. O valor do atributo da fila **QType** é QTLOC.

- As filas compartilhadas são suportadas apenas no z/OS

### **Fila de cluster**

Esta é uma fila física que armazena mensagens.. A fila existe no gerenciador de fila local ou em um ou mais dos gerenciadores de filas que pertencem ao mesmo cluster que o gerenciador de fila local..

Os aplicativos conectados ao gerenciador de fila local podem colocar mensagens em filas desse tipo, independentemente do local da fila Se uma instância da fila existir no gerenciador de filas locais, a fila se comporta da mesma maneira que uma fila local e os aplicativos conectados ao gerenciador de filas locais podem remover mensagens da fila. O valor do atributo da fila **QType** é QTCLUS.

### **Fila de alias**

Esta não é uma fila física-é um nome alternativo para uma fila local.. O nome da fila local para a qual o alias é resolvido faz parte da definição da fila de alias

Os aplicativos conectados ao gerenciador de fila local podem colocar mensagens e remover mensagens de filas de alias-as mensagens são colocadas e removidas da fila local na qual o alias é resolvido. O valor do atributo da fila **QType** é QTALS.

### **Fila remota**

Esta não é uma fila física-é a definição local de uma fila que existe em um gerenciador de filas remotas. A definição local da fila remota contém informações que informam ao gerenciador de filas locais como rotear mensagens para o gerenciador de filas remotas

Os aplicativos conectados ao gerenciador de filas locais podem colocar mensagens em filas remotas-as mensagens são colocadas na fila de transmissão local usada para rotear mensagens para o gerenciador de filas remotas Aplicativos não podem remover mensagens de filas remotas. O valor do atributo da fila **QType** é QTREM.

Uma definição de fila remota também pode ser usada para:

- Alias da fila de respostas

Nesse caso, o nome da definição é o nome de uma fila de resposta.. Para obter mais informações, consulte [Definições de alias da fila de resposta..](#)

- Alias do gerenciador de filas

Neste caso, o nome da definição é um alias para um gerenciador de filas e não o nome de uma fila. Para obter mais informações, consulte [Definições de alias do gerenciador de filas.](#)

### **Fila modelo**

Essa não é uma fila física-ela é um conjunto de atributos de fila a partir dos quais uma fila local pode ser criada

As mensagens não podem ser armazenadas em filas deste tipo

Alguns atributos de fila se aplicam a todos os tipos de fila; outros atributos de fila se aplicam apenas a determinados tipos de fila Os tipos de fila para os quais um atributo se aplica são indicados por um "X" em [Tabela 754 na página 1418](#) e tabelas subsequentes

[Tabela 754 na página 1418](#) resume os atributos específicos para filas. Os atributos são descritos em ordem alfabética

Os nomes dos atributos mostrados na tabela são os nomes usados com as chamadas MQINQ e MQSET. Quando comandos MQSC são usados para definir, alterar ou exibir atributos, nomes abreviados alternativos são usados; consulte [Comandos MQSC](#) para obter detalhes.

Na tabela a seguir, as colunas se aplicam da seguinte forma:

- A coluna para filas locais também se aplica a filas compartilhadas..
- A coluna para filas modelo indica quais atributos são herdados pela fila local criada da fila modelo.
- A coluna para filas de clusters indica os atributos que podem ser consultados quando a fila de clusters é aberta para consulta sozinha, ou para consulta e saída. Se a fila de clusters for aberta para consulta mais uma ou mais de entrada, navegação ou configuração, a coluna para filas locais se aplicará em vez disso..

<i>Tabela 754. Atributos para filas</i>						
<b>Atributo</b>	<b>Descrição</b>	<b>Local</b>	<b>Modelo</b>	<b>Alias</b>	<b>Remoto</b>	<b>Cluster</b>
<a href="#">AlterationDate</a>	Data em que a definição foi alterada pela última vez	X		X	X	
<a href="#">AlterationTime</a>	Horário em que a definição foi alterada pela última vez	X		X	X	
<a href="#">BackoutRequeueQName</a>	Nome excessivo da fila de reenfileiramento de restauração	X	X			
<a href="#">BackoutThreshold</a>	Limite de Recuperação	X	X			
<a href="#">BaseQName</a>	Nome da fila para o qual o alias é resolvido			X		
<a href="#">ClusterChannelNome</a>	Nome do canal do emissor de clusters	✓	✓			
<a href="#">ClusterName</a>	Nome do cluster ao qual a fila pertence	X		X	X	
<a href="#">ClusterNameList</a>	Nome do objeto da lista de nomes contendo nomes de clusters aos quais a fila pertence	X		X	X	
<a href="#">CreationDate</a>	Data em que a fila foi criada	X				
<a href="#">CreationTime</a>	Horário em que a fila foi criada	X				
<a href="#">CurrentQDepth</a>	Profundidade de fila atual	X				
<a href="#">DefBind</a>	Ligação padrão	X		X	X	X
<a href="#">DefinitionType</a>	Tipo de definição de fila	X	X			

Tabela 754. Atributos para filas (continuação)						
Atributo	Descrição	Local	Modelo	Alias	Remoto	Cluster
<a href="#">DefInputOpenOption</a>	Opções de abertura de entrada padrão	X	X			
<a href="#">DefPersistence</a>	Persistência padrão de mensagem	X	X	X	X	X
<a href="#">DefPriority</a>	Prioridade padrão da mensagem	X	X	X	X	X
<a href="#">DistLists</a>	Suporte à lista de distribuição	X	X			
<a href="#">HardenGetBackout</a>	Se deve-se manter uma contagem de restauração precisa	X	X			
<a href="#">InhibitGet</a>	Controla se as operações get para a fila são permitidas	X	X	X		
<a href="#">InhibitPut</a>	Controla se as operações put para a fila são permitidas	X	X	X	X	X
<a href="#">InitiationQName</a>	Nome da fila de inicialização	X	X			
<a href="#">MaxMsgLength</a>	Tamanho máximo da mensagem em bytes	X	X			
<a href="#">MaxQDepth</a>	Profundidade máxima da fila	X	X			
<a href="#">MediaLog</a>	Identidade da extensão de log mais antiga (ou receptor de diário mais antigo no IBM i) necessário para recuperação de mídia de uma fila especificada	✓	✓			
<a href="#">MsgDeliverySequence</a>	Seqüência de entrega de mensagens	X	X			
<a href="#">OpenInputCount</a>	Número de aberturas para entrada	X				
<a href="#">OpenOutputCount</a>	Número de aberturas para saída	X				
<a href="#">ProcessName</a>	Nome do processo	X	X			
<a href="#">QDepthHighEvent</a>	Controla se eventos de Alta Profundidade da Fila são gerados	X	X			
<a href="#">QDepthHighLimit</a>	Limite alto para profundidade da fila	X	X			

Tabela 754. Atributos para filas (continuação)

<b>Atributo</b>	<b>Descrição</b>	<b>Local</b>	<b>Modelo</b>	<b>Alias</b>	<b>Remoto</b>	<b>Cluster</b>
<u>QDepthLowEvent</u>	Controla se eventos de Baixa Profundidade da Fila são gerados	X	X			
<u>QDepthLowLimit</u>	Limite baixo para profundidade da fila	X	X			
<u>QDepthMaxEvent</u>	Controla se eventos de Fila Cheia são gerados	X	X			
<u>QDesc</u>	Descrição da fila	X	X	X	X	X
<u>QName</u>	Nome da fila	X		X	X	X
<u>QServiceInterval</u>	Destino para intervalo de serviço de fila	X	X			
<u>QServiceIntervalEvento</u>	Controla se os eventos Service Interval High ou Service Interval OK são gerados	X	X			
<u>QType</u>	Tipo de fila	X		X	X	X
<u>RemoteQMgrName</u>	Nome do gerenciador de fila remoto				X	
<u>RemoteQName</u>	Nome da fila remota				X	
<u>RetentionInterval</u>	Intervalo de retenção	X	X			
<u>Escopo</u>	Controla se uma entrada para a fila também existe em um diretório de célula	X		X	X	
<u>Compartilhamento</u>	Compartilhamento de fila	X	X			
<u>TriggerControl</u>	Controle de ativação	X	X			
<u>TriggerData</u>	Dados de Ativação	X	X			
<u>TriggerDepth</u>	Profundidade de ativação	X	X			
<u>TriggerMsgPriority</u>	Prioridade da mensagem limite para acionadores	X	X			
<u>TriggerType</u>	Tipo de ativação	X	X			
<u>Uso</u>	Uso da Fila	X	X			
<u>XmitQName</u>	Nome da fila de transmissão				X	



### **IBM i** *AlterationDate (sequência de caracteres de 12 bytes) em IBM i*

Data da última alteração da definição.

<i>Tabela 755. Tipos de fila aos quais este atributo se aplica</i>				
Local	Modelo	Alias	Remoto	Cluster
X		X	X	

Esta é a data em que a definição foi mudada pela última vez. O formato da data é YYYY-MM-DD, preenchido com dois espaços em branco finais para tornar o comprimento de 12 bytes (por exemplo, 1992-09-23--), em que -- representa dois caracteres em branco).

Os valores de determinados atributos (por exemplo, *CurrentQDepth*) mudam conforme o gerenciador de filas opera. As alterações nesses atributos não afetam *AlterationDate*.

Para determinar o valor desse atributo, use o seletor CAALTD com a chamada MQINQ.. O comprimento deste atributo é fornecido por LNDATE

### **IBM i** *AlterationTime (sequência de caracteres de 8 bytes) no IBM i*

Horário em que a definição foi mudada pela última vez

<i>Tabela 756. Tipos de fila aos quais este atributo se aplica</i>				
Local	Modelo	Alias	Remoto	Cluster
X		X	X	

Este é o horário em que a definição foi alterada pela última vez. O formato do horário é HH.MM.SS usando o relógio de 24 horas, com um zero à esquerda se a hora for menor que 10 (por exemplo 09.10.20). O horário é local.

Os valores de determinados atributos (por exemplo, *CurrentQDepth*) mudam conforme o gerenciador de filas opera. As alterações nesses atributos não afetam *AlterationTime*.

Para determinar o valor desse atributo, use o seletor CAALTT com a chamadas MQINQ. O comprimento desse atributo é fornecido por LNTIME

### **IBM i** *BackoutRequeueQName (sequência de caractere de 48 bytes) no IBM i .*

Nome da fila de reenfileiramento de restauração em excesso

<i>Tabela 757. Tipos de fila aos quais este atributo se aplica</i>				
Local	Modelo	Alias	Remoto	Cluster
X	X			

Os aplicativos em execução no WebSphere Application Server e aqueles que usam o IBM MQ Application Server Facilities usam esse atributo para determinar para onde as mensagens que foram restauradas devem ir. Para todos os outros aplicativos, além de permitir que seu valor seja consultado, o gerenciador de filas não toma nenhuma ação com base no valor do atributo.

Para determinar o valor desse atributo, use o seletor CABRQN com a chamada MQINQ.. O comprimento desse atributo é fornecido pelo LNQNN

### **IBM i** *BackoutThreshold (número inteiro assinado de 10 dígitos) em IBM i*

Limite de restauração.

<i>Tabela 758. Tipos de fila aos quais este atributo se aplica</i>				
<b>Local</b>	<b>Modelo</b>	<b>Alias</b>	<b>Remoto</b>	<b>Cluster</b>
X	X			

Aplicativos em execução no WebSphere Application Server e aqueles que usam o IBM MQ Application Server Facilities usam esse atributo para determinar se uma mensagem deve ser restaurada. Para todos os outros aplicativos, além de permitir que seu valor seja consultado, o gerenciador de filas não toma nenhuma ação com base no valor do atributo.

Para determinar o valor desse atributo, use o seletor IABTHR com a chamado MQINQ.

**IBM i BaseQName (sequência de caracteres de 48 bytes) no IBM i**

O nome da fila ao qual o alias é resolvido.

<i>Tabela 759. Tipos de fila aos quais este atributo se aplica</i>				
<b>Local</b>	<b>Modelo</b>	<b>Alias</b>	<b>Remoto</b>	<b>Cluster</b>
		X		

Este é o nome de uma fila definida para o gerenciador de fila local. (Para obter mais informações sobre nomes de filas, consulte a descrição do campo *ODON* no MQOD. A fila é um dos seguintes tipos:

**QTLOC**

Fila local.

**QTREM**

A definição local de uma fila remota

**QTCLUS**

Fila de clusters

Para determinar o valor desse atributo, use o seletor CABASQ com a chamada MQINQ.. O comprimento desse atributo é fornecido pelo LNQNN

**IBM i BaseType (estrutura de parâmetro de número inteiro) em IBM i .**

O tipo de objeto para o qual o alias é resolvido

<i>Tabela 760. Tipos de fila aos quais este atributo se aplica</i>				
<b>Local</b>	<b>Modelo</b>	<b>Alias</b>	<b>Remoto</b>	<b>Cluster</b>
		X		

Esse atributo pode ter um dos seguintes valores:

**OTQ**

O tipo de objeto base é uma fila

**OTTOP**

O tipo de objeto base é um tópico

**IBM i CFStrucName (sequência de caracteres de 12 bytes) no IBM i**

Nome da estrutura do recurso de acoplamento..


Tabela 761. Tipos de fila aos quais este atributo se aplica				
Local	Modelo	Alias	Remoto	Cluster
X	X			

Esse é o nome da estrutura do recurso de acoplamento em que as mensagens na fila são armazenadas. O primeiro caractere do nome está no intervalo de A a Z e os caracteres restantes estão no intervalo de A a Z, 0 a 9 ou em branco.

O nome completo da estrutura no recurso de acoplamento é obtido sufixando o valor do atributo do gerenciador de filas do **QSGName** com o valor do atributo da fila **CFStrucName**.

Esse atributo aplica-se apenas a filas compartilhadas; ele será ignorado se *QSGDisp* não tiver o valor QSGDSH.

Para determinar o valor desse atributo, use o seletor CACFSN com a chamada MQINQ.. O comprimento desse atributo é fornecido por LNCFSN.

 Esse atributo é suportado apenas no z/OS.

### **ClusterChannelNome (sequência de caracteres de 20 bytes)**

ClusterChannelNome é o nome genérico dos canais do emissor de clusters que usam essa fila como uma fila de transmissão. O atributo especifica quais canais de emissor de clusters enviam mensagens a um canal do receptor de clusters a partir desta fila de transmissão do cluster.

Tabela 762. Tipos de fila aos quais este atributo se aplica				
Local	Modelo	Alias	Remoto	Cluster
X	X			

A configuração padrão do gerenciador de filas é para todos os canais do emissor de clusters para enviar mensagens a partir de uma única fila de transmissão, SYSTEM.CLUSTER.TRANSMIT.QUEUE. Muda-se a configuração padrão modificando o atributo gerenciador de filas, **DefClusterXmitQueueType**. O valor padrão do atributo é SCTQ. É possível alterar o valor para CHANNEL. Ao configurar o atributo **DefClusterXmitQueueType** para CHANNEL, cada canal do emissor de cluster será padronizado para usar uma fila específica de transmissões de clusters, SYSTEM.CLUSTER.TRANSMIT.ChannelName.

É possível também configurar o atributo de fila de transmissão ClusterChannelNome para um canal do emissor de clusters manualmente. As mensagens que são destinadas ao gerenciador de filas conectado pelo canal do emissor de clusters são armazenadas na fila de transmissão que identifica o canal do emissor de clusters. Elas não são armazenadas na fila de transmissão do cluster padrão. Se você configurar o atributo ClusterChannelNome com espaços em branco, o canal alternará para a fila de transmissão do cluster padrão quando o canal for reiniciado. A fila padrão é de SYSTEM.CLUSTER.TRANSMIT.ChannelName ou SYSTEM.CLUSTER.TRANSMIT.QUEUE, dependendo do valor do atributo gerenciador de filas DefClusterXmitQueueType.

Ao especificar asteriscos, "\*", em **ClusterChannelNome**, é possível associar uma fila de transmissão a um conjunto de canais de emissor de clusters. Os asteriscos podem estar no início, fim ou qualquer número de locais no meio da sequência do nome de canal. **ClusterChannelNome** é limitado a 20 caracteres de comprimento: MQ\_CHANNEL\_NAME\_LENGTH.

### **ClusterName (sequência de caracteres de 48 bytes) no IBM i**

Nome do cluster ao qual a fila pertence..

<i>Tabela 763. Tipos de fila aos quais este atributo se aplica</i>				
<b>Local</b>	<b>Modelo</b>	<b>Alias</b>	<b>Remoto</b>	<b>Cluster</b>
X		X	X	

É o nome do cluster ao qual a fila pertence. Se a fila pertencer a mais de um cluster, *ClusterNameList* especifica o nome de um objeto de lista de nomes que identifica os clusters e *ClusterName* está em branco. Pelo menos um de *ClusterName* e *ClusterNameList* deve estar em branco

Para determinar o valor desse atributo, use o seletor CACLN com a chamada MQINQ.. O comprimento deste atributo é fornecido por LNCLUN

### **IBM i ClusterNameList (sequência de caracteres de 48 bytes) em IBM i**

Nome do objeto da lista de nomes contendo nomes de clusters aos quais a fila pertence.

<i>Tabela 764. Tipos de fila aos quais este atributo se aplica</i>				
<b>Local</b>	<b>Modelo</b>	<b>Alias</b>	<b>Remoto</b>	<b>Cluster</b>
X		X	X	

Esse é o nome de um objeto de lista de nomes que contém os nomes de clusters aos quais essa fila pertence. Se a fila pertencer a apenas um cluster, o objeto namelist contém apenas um nome. Como alternativa, *ClusterName* pode ser usado para especificar o nome do cluster, em cujo caso, *ClusterNameList* está em branco. Pelo menos um de *ClusterName* e *ClusterNameList* deve estar em branco

Para determinar o valor desse atributo, use o seletor CACLNL com a chamada MQINQ.. O comprimento deste atributo é fornecido por LNNLN

### **IBM i CreationDate (sequência de caractere de 12 bytes) no IBM i**

Data de criação da fila.

<i>Tabela 765. Tipos de fila aos quais este atributo se aplica</i>				
<b>Local</b>	<b>Modelo</b>	<b>Alias</b>	<b>Remoto</b>	<b>Cluster</b>
X				

Essa é a data em que a fila foi criada. O formato da data é YYYY-MM-DD, preenchido com dois espaços em branco finais para tornar o comprimento de 12 bytes (por exemplo, 1992-09-23--), em que -- representa dois caracteres em branco).

- No IBM i, a data de criação de uma fila pode ser diferente daquela da entidade do sistema operacional subjacente (arquivo ou espaço do usuário) que representa a fila..

Para determinar o valor desse atributo, use o seletor CACRTD com a chamada MQINQ. O comprimento deste atributo é fornecido por LNCRTD

### **IBM i CreationTime (sequência de caracteres de 8 bytes) em IBM i.**

A hora em que a fila foi criada

Tabela 766. Tipos de fila aos quais este atributo se aplica

Local	Modelo	Alias	Remoto	Cluster
X				

Este é o horário em que a fila foi criada O formato do horário é HH.MM.SS usando o relógio de 24 horas, com um zero à esquerda se a hora for menor que 10 (por exemplo 09.10.20). O horário é local.

- No IBM i, o horário de criação de uma fila pode ser diferente daquele da entidade do sistema operacional subjacente (arquivo ou espaço do usuário) que representa a fila..

Para determinar o valor desse atributo, use o seletor CACRTT com a chamada MQINQ.. O comprimento deste atributo é fornecido pelo LNCRTT

### IBM i **CurrentQDepth** (número inteiro assinado de 10 dígitos) no IBM i

Profundidade da fila atual

Tabela 767. Tipos de fila aos quais este atributo se aplica

Local	Modelo	Alias	Remoto	Cluster
X				

Esse é o número de mensagens atualmente na fila. Ele é incrementado durante uma chamada MQPUT e durante a restauração de uma chamada MQGET Ele é decrementado durante uma chamada MQGET não de navegação e durante a restauração de uma chamada MQPUT O efeito disso é que a contagem inclui mensagens que foram colocadas na fila dentro de uma unidade de trabalho, mas que ainda não foram confirmados, mesmo que eles não sejam elegíveis para serem recuperadas pela chamada MQGET Da mesma forma, ele exclui mensagens que foram recuperadas dentro de uma unidade de trabalho usando a chamada MQGET, mas que ainda precisam ser confirmadas..

A contagem também inclui mensagens que passaram seu tempo de expiração, mas ainda não foram descartadas, embora essas mensagens não sejam elegíveis para recuperação. Consulte o campo *MDEXP* descrito em “MQMD (Descritor de mensagens) em IBM i” na página 1148..

O processamento da unidade de trabalho e a segmentação de mensagens podem fazer com que o *CurrentQDepth* exceda *MaxQDepth*. No entanto, isso não afeta a capacidade de recuperação das mensagens- todas as mensagens na fila podem ser recuperadas usando a chamada MQGET da maneira normal

O valor desse atributo flutua conforme o gerenciador de filas opera.

Para determinar o valor desse atributo, use o seletor IACDEP com a chamada MQINQ..

### IBM i **DefBind** (número inteiro assinado de 10 dígitos) em IBM i

Ligação padrão..

Tabela 768. Tipos de fila aos quais este atributo se aplica

Local	Modelo	Alias	Remoto	Cluster
X		X	X	X

Esse atributo é a ligação padrão que é usada quando o OOBNDQ é especificado na chamada MQOPEN e a fila é uma fila de clusters DefBind pode ter um dos seguintes valores:

#### **BNDOPN**

Ligação fixa pela chamada MQOPEN.

## BNDNOT

Ligação não corrigida.

## BNDGRP

A ligação não é fixada pela chamada MQOPEN, mas é fixada em MQPUT para todas as mensagens em um grupo lógico.

Para determinar o valor deste atributo, use o seletor IADBND com a chamada MQINQ.

## **DefinitionType (número inteiro assinado de 10 dígitos) em IBM i**

O tipo de definição de fila.

<i>Tabela 769. Tipos de fila aos quais este atributo se aplica</i>				
Local	Modelo	Alias	Remoto	Cluster
X	X			

Isso indica como a fila foi definida O valor é um dos seguintes:

### QDPRE

Fila permanente predefinida..

A fila é uma fila permanente criada pelo administrador do sistema; somente ele pode excluí-la.

As filas predefinidas são criadas usando o comando MQSC DEFINE e podem ser excluídas apenas usando o comando MQSC DELETE . Filas predefinidas não podem ser criadas a partir de filas modelo.

Comandos podem ser emitidos por um operador ou por um usuário autorizado enviando uma mensagem de comando para a fila de entrada de comandos (consulte o atributo **CommandInputQName** descrito em [“Atributos para o gerenciador de filas no IBM i” na página 1449](#) ).

### QDPERM

Fila permanente definida dinamicamente..

A fila é uma fila permanente que foi criada por um aplicativo emitindo uma chamada MQOPEN com o nome de uma fila modelo especificada no descritor de objeto MQOD A definição de fila modelo tinha o valor QDPERM para o atributo **DefinitionType** .

Esse tipo de fila pode ser excluído usando a chamada MQCLOSE Consulte [“MQCLOSE \(Fechar objeto\) no IBM i” na página 1313](#) para obter mais detalhes.

O valor do atributo **QSGDisp** para uma fila dinâmica permanente é QSGDQM.

### QDTEMP

Fila temporária definida dinamicamente..

A fila é uma fila temporária que foi criada por um aplicativo emitindo uma chamada MQOPEN com o nome de uma fila modelo especificada no descritor de objeto MQOD.. A definição de fila modelo tinha o valor QDTEMP para o atributo **DefinitionType** .

Esse tipo de fila é excluído automaticamente pela chamada MQCLOSE quando ele é fechado pelo aplicativo que o criou

O valor do atributo **QSGDisp** para uma fila dinâmica temporária é QSGDQM

### QDSHAR

Fila compartilhada definida dinamicamente..

A fila é uma fila permanente compartilhada que foi criada por um aplicativo emitindo uma chamada MQOPEN com o nome de uma fila modelo especificada no MQOD do descritor de objeto A definição de fila modelo tinha o valor QDSHAR para o atributo **DefinitionType** .

Esse tipo de fila pode ser excluído usando a chamada MQCLOSE Consulte [“MQCLOSE \(Fechar objeto\) no IBM i” na página 1313](#) para obter mais detalhes.

O valor do atributo **QSGDisp** para uma fila dinâmica compartilhada é QSGDSH.

Este atributo em uma definição de fila modelo não indica como a fila modelo foi definida, porque as filas modelo são sempre predefinidas. Em vez disso, o valor desse atributo na fila modelo é usado para determinar o *DefinitionType* de cada uma das filas dinâmicas criadas a partir da definição de fila modelo usando a chamada MQOPEN.

Para determinar o valor deste atributo, use o seletor IADEFI com a chamada MQINQ.

**IBM i** **DefInputOpenOption (número inteiro assinado de 10 dígitos) em IBM i**  
Opção de abertura de entrada padrão

Tabela 770. Tipos de fila aos quais este atributo se aplica

Local	Modelo	Alias	Remoto	Cluster
X	X			

Esta é a maneira padrão na qual a fila deve ser aberta para entrada. Ele se aplicará se a opção OOINPQ for especificada na chamada MQOPEN quando a fila for aberta. Isso pode ter um dos seguintes valores:

**OOINPX**

Abra a fila para obter mensagens com acesso exclusivo.

A fila é aberta para uso com as chamadas MQGET subsequentes. A chamada falha com o código de razão RC2042 se a fila estiver aberta atualmente por este ou outro aplicativo para entrada de qualquer tipo (OOINPS ou OOINPX).

**OOINPS**

Abrir fila para obter mensagens com acesso compartilhado.

A fila é aberta para uso com as chamadas MQGET subsequentes. A chamada poderá ser bem-sucedida se a fila estiver atualmente aberta por este ou outro aplicativo com OOINPS, mas falhar com o código de razão RC2042 se a fila estiver atualmente aberta com OOINPX.

Para determinar o valor desse atributo, use o seletor IADINP com a chamada MQINQ.

**IBM i** **DefPersistence (número inteiro assinado de 10 dígitos) no IBM i**  
Persistência de mensagem padrão..

Tabela 771. Tipos de fila aos quais este atributo se aplica

Local	Modelo	Alias	Remoto	Cluster
X	X	X	X	X

Esta é a persistência padrão de mensagens na fila. Ele se aplicará se PEQDEF for especificado no descritor de mensagem quando a mensagem for colocada.

Se houver mais de uma definição no caminho de resolução do nome da fila, a persistência padrão será obtida do valor desse atributo na *primeira* definição no caminho no momento da chamada MQPUT ou MQPUT1. Isto poderia ser:

- uma fila de alias
- Uma fila local
- Uma definição local de uma fila remota
- Um alias do gerenciador de filas
- Uma fila de transmissão (por exemplo, a fila *DefXmitQName*)

Isso pode ter um dos seguintes valores:

#### PEPER

A mensagem é persistente

Isso significa que a mensagem sobrevive a falhas e reinicializações do sistema do gerenciador de fila. Mensagens persistentes não podem ser colocadas em:

- Filas dinâmicas temporárias
- Filas compartilhadas

Mensagens persistentes podem ser colocadas em filas dinâmicas permanentes e filas predefinidas.

#### PENPER

A mensagem não é persistente

Isso significa que a mensagem normalmente não sobrevive a falhas ou reinicializações do sistema do gerenciador de fila. Isso se aplicará mesmo se uma cópia intacta da mensagem for localizada no armazenamento auxiliar durante a reinicialização do gerenciador de filas.

No caso especial de filas compartilhadas, as mensagens não persistentes *não* sobrevivem a reinicializações de gerenciadores de fila no grupo de filas compartilhadas, mas não sobrevivem a falhas do recurso de acoplamento usado para armazenar mensagens nas filas compartilhadas.

As mensagens persistentes e não persistentes podem existir na mesma fila.

Para determinar o valor desse atributo, use o seletor IADPER com a chamada MQINQ.

#### **DefPriority (número inteiro assinado de 10 dígitos) no IBM i**

Prioridade da mensagem padrão.

Local	Modelo	Alias	Remoto	Cluster
X	X	X	X	X

Esta é a prioridade padrão para as mensagens na fila. Isso se aplicará se PRQDEF for especificado no descritor de mensagens quando a mensagem for colocada na fila.

Se houver mais de uma definição no caminho de resolução do nome da fila, a prioridade padrão para a mensagem será obtida do valor desse atributo na *primeira* definição no caminho no momento da operação put. Isto poderia ser:

- uma fila de alias
- Uma fila local
- Uma definição local de uma fila remota
- Um alias do gerenciador de filas
- Uma fila de transmissão (por exemplo, a fila *DefXmitQName*)

A maneira na qual uma mensagem é colocada em uma fila depende do valor do atributo

**MsgDeliverySequence** da fila:

- Se o atributo **MsgDeliverySequence** for MSPRIO, a posição lógica na qual uma mensagem é colocada na fila dependerá do valor do campo *MDPRI* no descritor de mensagens.
- Se o atributo **MsgDeliverySequence** for MSFIFO, as mensagens serão colocadas na fila como se tivessem uma prioridade igual ao *DefPriority* da fila resolvida independentemente do valor do campo *MDPRI* no descritor de mensagem. No entanto, o campo *MDPRI* retém o valor especificado pelo aplicativo que colocou a mensagem. Consulte o atributo **MsgDeliverySequence** descrito em “Atributos para filas” na página 1416 para obter mais informações.



As prioridades estão no intervalo de zero (menor) a *MaxPriority* (maior); consulte o atributo **MaxPriority** descrito em “Atributos para o gerenciador de filas no IBM i” na página 1449.

Para determinar o valor desse atributo, use o seletor IADPRI com a chamada MQINQ..

### **IBM i** **DefReadAhead (número inteiro assinado de 10 dígitos) on IBM i**

Especifica o comportamento de leitura antecipada padrão para mensagens não persistentes entregues para o cliente

*Tabela 773. Tipos de fila aos quais este atributo se aplica*

Local	Modelo	Alias	Remoto	Cluster
X	X	X		

DefReadAhead pode ser configurado para um dos seguintes valores:

#### **RAHNO**

Mensagens não persistentes não são enviadas para o cliente antes que um aplicativo as solicite. No máximo uma mensagem não persistente pode ser perdida se o cliente terminar de forma anormal.

#### **RAHYES**

As mensagens não persistentes são enviadas antecipadamente para o cliente antes que um aplicativo as solicite. As mensagens não persistentes poderão ser perdidas se o cliente for encerrado de forma anormal ou se o cliente não consumir todas as mensagens enviadas.

#### **RAHDIS**

Leia antes de mensagens não persistentes em não ativadas para esta fila. As mensagens não são enviadas antecipadamente para o cliente, independentemente de a leitura antecipada ser solicitada pelo aplicativo cliente.

Para determinar o valor desse atributo, use o seletor IADRAH com a chamada MQINQ..

### **IBM i** **DefPResp (número inteiro assinado de 10 dígitos) em IBM i**

O atributo default put response type (DEFPRESP) define o valor usado pelos aplicativos quando o Tipo PutResponso no MQPMO foi configurado como PMRASQ. Esse atributo é válido para todos os tipos de filas

*Tabela 774. Tipos de fila aos quais este atributo se aplica*

Local	Modelo	Alias	Remoto	Cluster
X	X	X	X	X

Isso pode ter um dos seguintes valores:

#### **SÍNC**

A operação put é emitida de forma síncrona retornando uma resposta.

#### **ASYN**

A operação put é emitida assincronamente, retornando um subconjunto de campos MQMD.

Para determinar o valor desse atributo, use o seletor IADPRT com a chamada MQINQ..

### **IBM i** **DistLists (número inteiro assinado de 10 dígitos) em IBM i**

Suporte à lista de distribuições

*Tabela 775. Tipos de fila aos quais este atributo se aplica*

Local	Modelo	Alias	Remoto	Cluster
X	X			

Indica se mensagens da lista de distribuição podem ser colocadas na fila. O atributo é configurado por um agente do canal de mensagens (MCA) para informar ao gerenciador de fila local se o gerenciador de filas na outra extremidade do canal suporta listas de distribuição.. Esse último gerenciador de filas (chamado de "gerenciador de filas de parceria") é aquele que em seguida recebe a mensagem, depois de ter sido removido da fila de transmissão local por um MCA de envio.

O atributo é configurado pelo MCA de envio sempre que ele estabelece uma conexão com o MCA de recebimento no gerenciador de filas de parceria Dessa forma, o MCA de envio pode fazer com que o gerenciador de filas locais coloque na fila de transmissão apenas as mensagens que o gerenciador de filas de parceria pode processar corretamente

Este atributo é principalmente para uso com filas de transmissão, mas o processamento descrito é executado independentemente do uso definido para a fila (consulte o atributo **Usage** ).

Isso pode ter um dos seguintes valores:

#### **DLSUPP**

Listas de distribuição suportadas.

Isso indica que as mensagens da lista de distribuição podem ser armazenadas na fila e transmitidas para o gerenciador de filas de parceria nesse formato Isso reduz a quantidade de processamento necessária para enviar a mensagem para vários destinos

#### **DLNSUP**

Listas de distribuição não suportadas.

Isso indica que as mensagens da lista de distribuição não podem ser armazenadas na fila, porque o gerenciador de filas de parceria não suporta listas de distribuição Se um aplicativo colocar uma mensagem da lista de distribuição e essa mensagem tiver que ser colocada nessa fila, o gerenciador de filas dividirá a mensagem da lista de distribuição e colocará as mensagens individuais na fila. Isso aumenta a quantia de processamento necessária para enviar a mensagem para vários destinos, mas assegura que as mensagens serão processadas corretamente pelo gerenciador de filas de parceria

Para determinar o valor desse atributo, use o seletor IADIST com a chamada MQINQ. Para alterar o valor desse atributo, use a chamada MQSET..

### **HardenGetRestauração (número inteiro assinado de 10 dígitos) em IBM i**

Se deve manter uma contagem de restauração precisa.

*Tabela 776. Tipos de fila aos quais este atributo se aplica*

<b>Local</b>	<b>Modelo</b>	<b>Alias</b>	<b>Remoto</b>	<b>Cluster</b>
X	X			

Para cada mensagem, uma contagem é mantida do número de vezes que a mensagem é recuperada por uma chamada MQGET em uma unidade de trabalho e essa unidade de trabalho posteriormente restaurada. Essa contagem está disponível no campo *MDBOC* no descritor de mensagens após a chamada MQGET ter sido concluída

A contagem de restaurações de mensagens sobrevive quando o gerenciador de filas é reiniciado No entanto, para assegurar que a contagem seja precisa, as informações devem ser "reforçadas" (registradas no disco ou em outro dispositivo de armazenamento permanente) cada vez que uma mensagem for recuperada por uma chamada MQGET dentro de uma unidade de trabalho para essa fila. Se isso não for feito, e uma falha do gerenciador de filas ocorrer junto com a restauração da chamada MQGET, a contagem poderá não ser incrementada

O reforço de informações para cada chamada MQGET dentro de uma unidade de trabalho, no entanto, impõe um custo de desempenho e o atributo **HardenGetBackout** deve ser configurado como QABH somente se a contagem tiver que ser precisa

- No IBM i, a contagem de restaurações de mensagens é sempre reforçada, independentemente da configuração desse atributo

Os valores a seguir são possíveis:

#### **QABH**

Contagem de restauração lembrada.

O reforço é usado para assegurar que a contagem de restauração para mensagens nesta fila seja precisa.

#### **QABNH**

A contagem de restaurações pode não ser lembrada

O reforço não é usado para assegurar que a contagem de restauração para mensagens nesta fila seja precisa. Portanto, a contagem pode ser menor do que deveria ser.

Para determinar o valor deste atributo, use o seletor IAHGB com a chamada MQINQ.

### **IBM i InhibitGet (número inteiro assinado de 10 dígitos) no IBM i**

Controla se operações get para esta fila são permitidas.

*Tabela 777. Tipos de fila aos quais este atributo se aplica*

Local	Modelo	Alias	Remoto	Cluster
X	X	X		

Se a fila for uma fila de alias, as operações get devem ser permitidas para o alias e a fila base no momento da operação get, para que a chamada MQGET seja bem-sucedida. O valor é um dos seguintes:

#### **QAGETI**

As operações de obtenção são inibidas

Chamadas MQGET falham com o código de razão RC2016. Isso inclui chamadas MQGET que especificam GMBRWF ou GMBRWN..

**Nota:** Se uma chamada MQGET operando em uma unidade de trabalho for concluída com êxito, alterar o valor do atributo **InhibitGet** após para QAGETI não evitará que a unidade de trabalho seja confirmada.

#### **QAGETA**

As operações de obtenção são permitidas

Para determinar o valor desse atributo, use o seletor IAIGET com a chamada MQINQ. Para alterar o valor desse atributo, use a chamada MQSET..

### **IBM i InhibitPut (número inteiro assinado de 10 dígitos) no IBM i**

Controla se operações put para esta fila são permitidas.

*Tabela 778. Tipos de fila aos quais este atributo se aplica*

Local	Modelo	Alias	Remoto	Cluster
X	X	X	X	X

Se houver mais de uma definição no caminho de resolução de nome da fila, as operações put deverão ser permitidas para *cada* definição no caminho (incluindo quaisquer definições de alias do gerenciador de filas) no momento da operação put, para que a chamada MQPUT ou MQPUT1 seja bem-sucedida. Isso pode ter um dos seguintes valores:

## QAPUTI

Operações put são inibidas.

Chamadas MQPUT e MQPUT1 falham com o código de razão RC2051.

**Nota:** Se uma chamada MQPUT operando em uma unidade de trabalho for concluída com êxito, mudar o valor do atributo **InhibitPut** posteriormente para QAPUTI não evitará que a unidade de trabalho seja confirmada.

## QAPUTA

Operações put são permitidas.

Para determinar o valor desse atributo, use o seletor IAIPUT com a chamada MQINQ.. Para alterar o valor desse atributo, use a chamada MQSET..

## IBM i **InitiationQName** (sequência de caracteres de 48 bytes) no IBM i

Nome da fila de iniciação.

*Tabela 779. Tipos de fila aos quais este atributo se aplica*

Local	Modelo	Alias	Remoto	Cluster
X	X			

Este é o nome de uma fila definida no gerenciador de fila local; a fila deve ser do tipo QTLOC O gerenciador de filas envia uma mensagem do acionador para a fila de inicialização quando a inicialização do aplicativo é necessária como resultado de uma mensagem que chega na fila à qual esse atributo pertence A fila de inicialização deve ser monitorada por um aplicativo monitor acionador que iniciará o aplicativo apropriado após o recebimento da mensagem do acionador.

Para determinar o valor deste atributo, use o seletor CAINIQ com a chamada MQINQ. O comprimento desse atributo é fornecido pelo LNQNN

## IBM i **MaxMsgComprimento** (número inteiro assinado de 10 dígitos) em IBM i

Comprimento máximo da mensagem, em bytes.

*Tabela 780. Tipos de fila aos quais este atributo se aplica*

Local	Modelo	Alias	Remoto	Cluster
X	X			

Este é um limite superior para o comprimento da mensagem *física* mais longa que pode ser colocada na fila No entanto, como o atributo da fila **MaxMsgLength** pode ser configurado independentemente do atributo do gerenciador de filas **MaxMsgLength** , o limite superior real para o comprimento da mensagem física mais longa que pode ser colocada na fila é o menor desses dois valores

Se o gerenciador de filas suportar a segmentação, será possível para um aplicativo colocar uma mensagem *lógica* que seja maior que o menor dos dois atributos **MaxMsgLength** , mas somente se o aplicativo especificar o sinalizador MFSEGA no MQMD Se esse sinalizador for especificado, o limite superior para o comprimento de uma mensagem lógica será 999 999 999 bytes, mas, geralmente, restrições de recursos impostas pelo sistema operacional ou pelo ambiente no qual o aplicativo está em execução, resultará em um limite inferior.

Uma tentativa de colocar na fila uma mensagem muito longa falha com o código de razão:

- RC2030 se a mensagem for muito grande para a fila
- RC2031 se a mensagem for muito grande para o gerenciador de filas, mas não muito grande para a fila

O limite inferior para o atributo **MaxMsgLength** é zero. O limite superior é determinado pelo ambiente:

- No IBM i, o comprimento máximo da mensagem é 100 MB (104 857 600 bytes).

Para obter mais informações, consulte o parâmetro **BUFLEN** descrito em “MQPUT (mensagem Put) em IBM i” na página 1380

Para determinar o valor deste atributo, use o seletor IAMLEN com a chamada MQINQ.

### **IBM i** **MaxQDepth (número inteiro assinado de 10 dígitos) no IBM i**

Profundidade máxima da fila.

*Tabela 781. Tipos de fila aos quais este atributo se aplica*

Local	Modelo	Alias	Remoto	Cluster
X	X			

Este é o limite superior definido para o número de mensagens físicas que podem existir na fila a qualquer momento. Uma tentativa de colocar uma mensagem em uma fila que já contém mensagens *MaxQDepth* falha com o código de razão RC2053.

O processamento da unidade de trabalho e a segmentação de mensagens podem fazer o número real de mensagens físicas na fila exceder *MaxQDepth*. No entanto, isso não afeta a capacidade de recuperação das mensagens- todas as mensagens na fila podem ser recuperadas usando a chamada MQGET da maneira normal.

O valor deste atributo é zero ou maior. O limite superior é determinado pelo ambiente..

**Nota:** É possível que o espaço de armazenamento disponível para a fila seja esgotado, mesmo se houver menos de *MaxQDepth* mensagens na fila.

Para determinar o valor desse atributo, use o seletor IAMDEP com a chamada MQINQ..

### **IBM i** **MediaLog (número inteiro assinado de 10 dígitos) no IBM i**

Identidade da extensão de log (ou receptor de diário em IBM i) necessário para a recuperação de mídia de uma fila específica.

*Tabela 782. Tipos de fila aos quais este atributo se aplica*

Local	Modelo	Alias	Remoto	Cluster
X	X			

Nos gerenciadores de filas em que a criação de log circular está em uso, o valor é retornado como uma cadeia nula.

### **IBM i** **MsgDeliverySequência (número inteiro assinado de 10 dígitos) em IBM i**

Sequência de entrega de mensagens.

*Tabela 783. Tipos de fila aos quais este atributo se aplica*

Local	Modelo	Alias	Remoto	Cluster
X	X			

Isso determina a ordem na qual as mensagens são retornadas ao aplicativo pela chamada MQGET:

#### **MSFIFO**

Mensagens são retornadas na ordem FIFO (primeiro a entrar, primeiro a sair).

Isso significa que uma chamada MQGET retornará a *primeira* mensagem que satisfaz os critérios de seleção especificados na chamada, independentemente da prioridade da mensagem.

### **MSPRIO**

As mensagens são retornadas em ordem de prioridade

Isso significa que uma chamada MQGET retornará a mensagem *de prioridade mais alta* que satisfaça os critérios de seleção especificados na chamada. Dentro de cada nível de prioridade, as mensagens são retornadas na ordem FIFO (primeiro a entrar, primeiro a sair)

Se os atributos relevantes forem alterados enquanto houver mensagens na fila, a sequência de entrega será a seguinte:

- A ordem na qual as mensagens são retornadas pela chamada MQGET é determinada pelos valores dos atributos **MsgDeliverySequence** e **DefPriority** em vigor para a fila no momento em que a mensagem chega na fila:
  - Se *MsgDeliverySequence* for MSFIFO quando a mensagem chegar, a mensagem será colocada na fila como se sua prioridade fosse *DefPriority*. Isso não afeta o valor do campo *MDPRI* no descritor de mensagens da mensagem; esse campo retém o valor que ele tinha quando a mensagem foi colocada pela primeira vez
  - Se *MsgDeliverySequence* for MSPRIO quando a mensagem chegar, a mensagem será colocada na fila no local apropriado para a prioridade fornecida pelo campo *MDPRI* no descritor de mensagens

Se o valor do atributo **MsgDeliverySequence** for alterado enquanto houver mensagens na fila, a ordem das mensagens na fila não será alterada.

Se o valor do atributo **DefPriority** for alterado enquanto houver mensagens na fila, as mensagens não serão necessariamente entregues na ordem FIFO, mesmo que o atributo **MsgDeliverySequence** esteja configurado como MSFIFO; aqueles que foram colocados na fila na prioridade mais alta são entregues primeiro.

Para determinar o valor desse atributo, use o seletor IAMDS com a chamadas MQINQ.

**IBM i** **OpenInputContagem (número inteiro assinado de 10 dígitos) em IBM i**  
Número de aberturas para entrada.

Tabela 784. Tipos de fila aos quais este atributo se aplica

Local	Modelo	Alias	Remoto	Cluster
X				

Este é o número de identificadores que são atualmente válidos para remover mensagens da fila com a chamada MQGET. É o número total desses identificadores conhecidos no gerenciador de filas *local*. Se a fila for uma fila compartilhada, a contagem não incluirá aberturas para entrada que foram executadas para a fila em outros gerenciadores de fila no grupo de filas compartilhadas ao qual o gerenciador de fila local pertence..

A contagem inclui identificadores em que uma fila de alias que é resolvida para essa fila foi aberta para entrada. A contagem não inclui manipulações nas quais a fila foi aberta para ações que não incluíam entrada (por exemplo, uma fila aberta apenas para procura)...

O valor desse atributo flutua conforme o gerenciador de filas opera.

Para determinar o valor desse atributo, use o seletor IAOIC com a chamada MQINQ..

**IBM i** **OpenOutputContagem (número inteiro assinado de 10 dígitos) em IBM i**  
Número de aberturas para saída.

Tabela 785. Tipos de fila aos quais este atributo se aplica

Local	Modelo	Alias	Remoto	Cluster
X				

Esse é o número de identificadores que são válidos atualmente para incluir mensagens na fila com a chamada MQPUT. É o número total desses identificadores conhecidos do gerenciador de filas *local*; ele não inclui aberturas para saída que foram executadas para essa fila em gerenciadores de filas remotas. Se a fila for uma fila compartilhada, a contagem não incluirá aberturas para a saída que foram executadas para a fila em outros gerenciadores de filas no grupo de filas compartilhadas ao qual o gerenciador de filas locais pertence.

A contagem inclui identificadores nos quais uma fila de alias resolvida para essa fila foi aberta para saída. A contagem não inclui identificadores nos quais a fila foi aberta para ações que não incluíam saída (por exemplo, uma fila aberta apenas para consulta)...

O valor desse atributo flutua conforme o gerenciador de filas opera.

Para determinar o valor desse atributo, use o seletor IA00C com a chamada MQINQ..

### **IBM i** *ProcessName (sequência de 48 bytes) no IBM i*

Nome do processo.

Tabela 786. Tipos de fila aos quais este atributo se aplica

Local	Modelo	Alias	Remoto	Cluster
X	X			

Este é o nome de um objeto de processo definido no gerenciador de fila local. O objeto de processo identifica um programa que pode atender a fila..

Para determinar o valor deste atributo, use o seletor CAPRON com a chamada MQINQ. O comprimento deste atributo é fornecido por LNPRON

### **IBM i** *QDepthHighEvento (número inteiro assinado de 10 dígitos) no IBM i*

Controla se os eventos de Alta Profundidade da Fila são gerados

Tabela 787. Tipos de fila aos quais este atributo se aplica

Local	Modelo	Alias	Remoto	Cluster
X	X			

Um evento de Profundidade da Fila Alta indica que um aplicativo colocou uma mensagem em uma fila, o que fez com que o número de mensagens na fila se tornasse maior ou igual ao limite alto de profundidade da fila (consulte o atributo **QDepthHighLimit** )...

**Nota:** O valor desse atributo pode mudar dinamicamente.

QDepthHighO evento pode ter um de dois valores:

#### **EVRDIS**

Relatório de eventos desativado.

#### **EVRENA**

Relatório de eventos ativado

Para obter mais informações sobre os eventos, consulte [Monitoramento de eventos](#)

Para determinar o valor desse atributo, use o seletor IAQDHE com a chamada MQINQ..

**IBM i** **QDepthHighLimite (número inteiro assinado de 10 dígitos) em IBM i**  
Limite alto para a profundidade da fila

*Tabela 788. Tipos de fila aos quais este atributo se aplica*

Local	Modelo	Alias	Remoto	Cluster
X	X			

Este é o limite com relação ao qual a profundidade da fila é comparada para gerar um evento de Alta Profundidade da fila... Este evento indica que um aplicativo colocou uma mensagem em uma fila e isso fez com que o número de mensagens na fila se tornasse maior ou igual ao limite alto de profundidade da fila. Consulte o atributo **QDepthHighEvent**

O valor é expresso como uma porcentagem da profundidade máxima da fila (atributo **MaxQDepth**) e está no intervalo de zero a 100. O valor padrão é 80.

Para determinar o valor desse atributo, use o seletor IAQDHL com a chamada MQINQ..

**IBM i** **QDepthLowEvento (número inteiro assinado de 10 dígitos) em IBM i**  
Controla se os eventos de Baixa Profundidade da Fila são gerados

*Tabela 789. Tipos de fila aos quais este atributo se aplica*

Local	Modelo	Alias	Remoto	Cluster
X	X			

Um evento de Baixa Profundidade da Fila indica que um aplicativo recuperou uma mensagem de uma fila, o que fez com que o número de mensagens na fila se torne menor ou igual ao limite baixo de profundidade da fila (consulte o atributo **QDepthLowLimit**).

**Nota:** O valor desse atributo pode mudar dinamicamente.

QDepthLowO evento pode ter um dos seguintes valores:

**EVRDIS**

Relatório de eventos desativado.

**EVRENA**

Relatório de eventos ativado

Para obter mais informações sobre os eventos, consulte [Monitoramento de eventos](#)

Para determinar o valor desse atributo, use o seletor IAQDLE com a chamada MQINQ..

**IBM i** **QDepthLowLimite (número inteiro assinado de 10 dígitos) em IBM i**  
Limite baixo para a profundidade da fila

*Tabela 790. Tipos de fila aos quais este atributo se aplica*

Local	Modelo	Alias	Remoto	Cluster
X	X			

Este é o limite com relação ao qual a profundidade da fila é comparada para gerar um evento de Baixa Profundidade da fila... Esse evento indica que um aplicativo recuperou uma mensagem de uma fila e isso



fez com que o número de mensagens na fila se torne menor ou igual ao limite baixo de profundidade da fila. Consulte o atributo **QDepthLowEvent**

O valor é expresso como uma porcentagem da profundidade máxima da fila (atributo **MaxQDepth**) e está no intervalo de zero a 100. O valor padrão é 20.

Para determinar o valor desse atributo, use o seletor IAQDLL com a chamada MQINQ..

### **IBM i** **QDepthMaxEvento (número inteiro assinado de 10 dígitos) em IBM i**

Controla se os eventos de Fila Cheia são gerados

<i>Tabela 791. Tipos de fila aos quais este atributo se aplica</i>				
<b>Local</b>	<b>Modelo</b>	<b>Alias</b>	<b>Remoto</b>	<b>Cluster</b>
X	X			

Um evento Fila Cheia indica que um put para uma fila foi rejeitado porque a fila está cheia, ou seja, a profundidade da fila já atingiu seu valor máximo.

**Nota:** O valor desse atributo pode mudar dinamicamente.

Isso pode ter um dos seguintes valores:

#### **EVRDIS**

Relatório de eventos desativado.

#### **EVRENA**

Relatório de eventos ativado

Para obter mais informações sobre os eventos, consulte [Monitoramento de eventos](#)

Para determinar o valor deste atributo, use o seletor IAQDME com a chamada MQINQ..

### **IBM i** **QDesc (sequência de caracteres de 64 bytes) no IBM i**

Descrição da fila..

<i>Tabela 792. Tipos de fila aos quais este atributo se aplica</i>				
<b>Local</b>	<b>Modelo</b>	<b>Alias</b>	<b>Remoto</b>	<b>Cluster</b>
X	X	X	X	X

Este é um campo que pode ser usado para comentário descritivo.. O conteúdo do campo não é significativo para o gerenciador de filas, mas o gerenciador de filas pode requerer que o campo contenha apenas caracteres que possam ser exibidos.. Ele não pode conter nenhum caractere nulo; se necessário, será preenchido à direita com espaços em branco. Em uma instalação DBCS, o campo pode conter caracteres DBCS (sujeito a um comprimento máximo de campo de 64 bytes).

**Nota:** Se esse campo contiver caracteres que não estão no conjunto de caracteres do gerenciador de filas (conforme definido pelo atributo do gerenciador de filas **CodedCharSetId**), esses caracteres poderão ser convertidos incorretamente se esse campo for enviado para outro gerenciador de filas..

Para determinar o valor desse atributo, use o seletor CAQD com a chamada MQINQ.. O comprimento desse atributo é fornecido pelo LNQD

### **IBM i** **QName (sequência de caracteres de 48 bytes) em IBM i .**

Nome da Fila.

Tabela 793. Tipos de fila aos quais este atributo se aplica

Local	Modelo	Alias	Remoto	Cluster
X		X	X	X

Este é o nome de uma fila definida no Gerenciador de Filas Local. Para obter mais informações sobre nomes de filas, consulte Regras para nomear objetos IBM MQ. Todas as filas definidas em um gerenciador de filas compartilham o mesmo namespace da fila. Portanto, uma fila QTLOC e uma fila QTALS não podem ter o mesmo nome.

Para determinar o valor desse atributo, use o seletor CAQN com a chamada MQINQ.. O comprimento desse atributo é fornecido pelo LNQNN

**IBM i** **QServiceInterval (número inteiro assinado de 10 dígitos) no IBM i**

Destino para intervalo de serviço de fila.

Tabela 794. Tipos de fila aos quais este atributo se aplica

Local	Modelo	Alias	Remoto	Cluster
X	X			

Este é o intervalo de serviço usado para comparação para gerar eventos de Intervalo de Serviço Alto e de Intervalo de Serviço OK. Consulte o atributo **QServiceIntervalEvent**

O valor está em unidades de milissegundos e está no intervalo de zero a 999 999 999.

Para determinar o valor desse atributo, use o seletor IAQSI com a chamada MQINQ..

**IBM i** **QServiceIntervalEvento (número inteiro assinado de 10 dígitos) em IBM i**

Controla se os eventos Service Interval High ou Service Interval OK são gerados.

Tabela 795. Tipos de fila aos quais este atributo se aplica

Local	Modelo	Alias	Remoto	Cluster
X	X			

- Um evento Alto Intervalo de Serviço é gerado quando uma verificação indica que nenhuma mensagem foi recuperada da fila pelo menos pelo tempo indicado pelo atributo **QServiceInterval**.
- Um evento Service Interval OK é gerado quando uma verificação indica que as mensagens foram recuperadas da fila dentro do tempo indicado pelo atributo **QServiceInterval**.

**Nota:** O valor desse atributo pode mudar dinamicamente.

Esse atributo pode ter um dos seguintes valores:

**QSIEHI**

Eventos de Alto Intervalo de Serviço da Fila ativados.

- Eventos de Alto Intervalo de Serviço da Fila são **ativados** e
- Os eventos OK do Intervalo de Serviço da Fila são **desativados**.

**QSIEOK**

Eventos de intervalo de serviço de fila OK ativados.

- Os eventos Alto Intervalo de Serviço da Fila são **desativados** e
- Eventos OK do Intervalo de Serviço da Fila são **ativados**.

## QSIENO

Nenhum evento de intervalo de serviço de filas ativado

- Os eventos Alto Intervalo de Serviço da Fila são **desativados** e
- Os eventos de Intervalo de Serviço de Fila OK também são **desativados**.

Para filas compartilhadas, o valor deste atributo é ignorado; o valor QSIENO é assumido.

Para obter mais informações sobre os eventos, consulte [Monitoramento de eventos](#)

Para determinar o valor desse atributo, use o seletor IAQSIE com a chamadas MQINQ.

## **QSGDisp (número inteiro assinado de 10 dígitos) em IBM i**

Disposição do grupo de compartilhamento de filas

<i>Tabela 796. Tipos de fila aos quais este atributo se aplica</i>				
Local	Modelo	Alias	Remoto	Cluster
X		X	X	

Isso especifica a disposição da fila O valor é um dos seguintes:

### QSGDQM

Disposição do gerenciador de filas..

O objeto tem a disposição do gerenciador de filas Isso significa que a definição de objeto é conhecida apenas para o gerenciador de filas locais; a definição não é conhecida para outros gerenciadores de filas no grupo de filas compartilhadas

É possível para cada gerenciador de filas no grupo de filas compartilhadas ter um objeto com o mesmo nome e tipo que o objeto atual, mas eles são objetos separados e não há correlação entre eles.. Seus atributos não são restritos a serem iguais entre si.

### QSGDCP

Copiado-disposição do objeto

O objeto é uma cópia local de uma definição de objeto principal que existe no repositório compartilhado. Cada gerenciador de filas no grupo de filas compartilhadas pode ter a sua própria cópia do objeto. Inicialmente, todas as cópias têm os mesmos atributos, mas usando comandos MQSC, cada cópia pode ser alterada para que seus atributos sejam diferentes daqueles das outras cópias.. Os atributos das cópias são ressincronizados quando a definição principal no repositório compartilhado é alterado..

### QSGDSH

Disposição compartilhada..

O objeto possui uma disposição compartilhada Isso significa que existe no repositório compartilhado uma única instância do objeto conhecido por todos os gerenciadores de filas no grupo de filas compartilhadas. Quando um gerenciador de filas no grupo acessa o objeto, ele acessa a única instância compartilhada do objeto

Para determinar o valor desse atributo, use o seletor IAQSGD com a chamada MQINQ..

 Esse atributo é suportado apenas no z/OS

## **QType (número inteiro assinado de 10 dígitos) em IBM i**

Tipo de fila.

Tabela 797. Tipos de fila aos quais este atributo se aplica				
Local	Modelo	Alias	Remoto	Cluster
X		X	X	X

Este atributo possui um dos seguintes valores:

**QTALS**

Definição de fila de alias

**QTCLUS**

Fila de clusters

**QTLOC**

Fila local.

**QTREM**

A definição local de uma fila remota

Para determinar o valor desse atributo, use o seletor IAQTYP com a chamada MQINQ..

**IBM i RemoteQMgrNome (sequência de caractere de 48 bytes) no IBM i .**

Nome do gerenciador de filas remoto.

Tabela 798. Tipos de fila aos quais este atributo se aplica				
Local	Modelo	Alias	Remoto	Cluster
			X	

Este é o nome do gerenciador de filas remotas no qual a fila *RemoteQName* está definida Se a fila *RemoteQName* tiver um valor *QSGDisp* de *QSGDCP* ou *QSGDSH*, *RemoteQMgrName* poderá ser o nome do grupo de filas compartilhadas que possui *RemoteQName*.

Se um aplicativo abrir a definição local de uma fila remota, *RemoteQMgrName* não deverá estar em branco e não deverá ser o nome do gerenciador de fila local Se *XmitQName* estiver em branco, a fila local com o mesmo nome de *RemoteQMgrName* será usada como a fila de transmissão Se não houver nenhuma fila com o nome *RemoteQMgrName*, a fila identificada pelo atributo do gerenciador de filas **DefXmitQName** será usada

Se essa definição for usada para um alias do gerenciador de filas, *RemoteQMgrName* será o nome do gerenciador de filas que está sendo alias.... Ele pode ser o nome do gerenciador de fila local Caso contrário, se *XmitQName* estiver em branco quando a abertura ocorrer, deverá haver uma fila local com o mesmo nome que *RemoteQMgrName* ; esta fila é usada como a fila de transmissão

Se essa definição for usada para um alias de resposta, esse nome será o nome do gerenciador de fila que deve ser o *MDRM*

**Nota:** Nenhuma validação é executada no valor especificado para este atributo quando a definição de fila é criada ou modificada..

Para determinar o valor desse atributo, use o seletor CARQMN com a chamadas MQINQ. O comprimento desse atributo é fornecido por LNQMNM

**IBM i RemoteQName (sequência de caracteres de 48 bytes) em IBM i**

Nome da fila remota..

Tabela 799. Tipos de fila aos quais este atributo se aplica				
Local	Modelo	Alias	Remoto	Cluster
			X	

Este é o nome da fila como ele é conhecido no gerenciador de filas remotas *RemoteQMGrName*

Se um aplicativo abrir a definição local de uma fila remota, quando a abertura ocorrer *RemoteQName* não deverá ficar em branco.

Se essa definição for usada para uma definição de alias do gerenciador de fila, quando a abertura ocorrer, *RemoteQName* deverá estar em branco

Se a definição for usada para um alias de resposta, esse nome será o nome da fila que deve ser o *MDRQ*

**Nota:** Nenhuma validação é executada no valor especificado para este atributo quando a definição de fila é criada ou modificada..

Para determinar o valor desse atributo, use o seletor CARQN com a chamada MQINQ.. O comprimento desse atributo é fornecido pelo LNQNN

### **IBM i** *RetentionInterval (número inteiro assinado de 10 dígitos) em IBM i*

Intervalo de retenção.

Tabela 800. Tipos de fila aos quais este atributo se aplica				
Local	Modelo	Alias	Remoto	Cluster
X	X			

Este é o horário em que a fila deve ser retida Após esse tempo, a fila será elegível para exclusão.

O tempo é medido em horas contando a partir da data e hora em que a fila foi criada. A data de criação da fila é gravada no *CreationDate* e o horário de criação da fila é registrado no atributo **CreationTime**

Essas informações são fornecidas para permitir que um aplicativo de manutenção ou o operador identifique e exclua filas que não são mais necessárias.

**Nota:** O gerenciador de filas nunca tenta excluir filas com base nesse atributo ou evitar a exclusão de filas com um intervalo de retenção que não expirou; é responsabilidade do usuário fazer com que qualquer ação necessária seja executada.

Um intervalo de retenção realista deve ser usado para evitar o acúmulo de filas dinâmicas permanentes (consulte *DefinitionType*). No entanto, esse atributo também pode ser usado com filas predefinidas

Para determinar o valor desse atributo, use o seletor IARINT com a chamada MQINQ.

### **IBM i** *Escopo (número inteiro assinado de 10 dígitos) em IBM i*

Controla se uma entrada para esta fila também existe em um diretório de células.

Tabela 801. Tipos de fila aos quais este atributo se aplica				
Local	Modelo	Alias	Remoto	Cluster
X		X	X	

Um diretório de célula é fornecido por um serviço de Nome instalável Isso pode ter um dos seguintes valores:

#### **SCOQM**

escopo do gerenciador de filas.

A definição de fila possui o escopo do gerenciador de filas Isso significa que a definição da fila não se estende além do gerenciador de filas que possui. Para abrir a fila para saída de algum outro gerenciador de filas, o nome do gerenciador de filas proprietário deve ser especificado ou o outro gerenciador de filas deve ter uma definição local da fila.

### SCOCEL

Escopo da célula..

A definição de fila possui o escopo da célula Isso significa que a definição de fila também é colocada em um diretório de célula disponível para todos os gerenciadores de fila na célula A fila pode ser aberta para saída de qualquer um dos gerenciadores de filas na célula apenas especificando o nome da fila; o nome do gerenciador de filas que possui a fila não precisa ser especificado No entanto, a definição de fila não está disponível para qualquer gerenciador de fila na célula que também possui uma definição local de uma fila com esse nome, pois a definição local tem precedência..

Um diretório de célula é fornecido por um serviço de nome instalável, como LDAP (Lightweight Directory Access Protocol) Observe que o IBM MQ não suporta mais o serviço de nomes DCE (Distributed Computing Environment) que foi usado anteriormente para inserir definições de filas em um diretório DCE (também não mais suportado).

O modelo e as filas dinâmicas não podem ter escopo de célula

Este valor será válido apenas se um serviço de nomes que suporta um diretório de célula tiver sido configurado..

Para determinar o valor desse atributo, use o seletor IASCOP com a chamada MQINQ..

O suporte para esse atributo está sujeito às seguintes restrições:

- No IBM i, o atributo é suportado mas apenas o SCOQM é válido

### **Compartilhabilidade (número inteiro assinado de 10 dígitos) em IBM i**

Se a fila pode ser compartilhada para a entrada

<i>Tabela 802. Tipos de fila aos quais este atributo se aplica</i>				
Local	Modelo	Alias	Remoto	Cluster
X	X			

Isso indica se a fila pode ser aberta para entrada várias vezes simultaneamente. Isso pode ter um dos seguintes valores:

#### QASHR

A fila é compartilhável

Várias aberturas com a opção OOINPS são permitidas

#### QANSHR

A fila não é compartilhável

Uma chamada MQOPEN com a opção OOINPS é tratada como OINPX.

Para determinar o valor desse atributo, use o seletor IASHAR com a chamada MQINQ..

### **TriggerControl (número inteiro assinado de 10 dígitos) no IBM i**

Acionador de controle.

<i>Tabela 803. Tipos de fila aos quais este atributo se aplica</i>				
<b>Local</b>	<b>Modelo</b>	<b>Alias</b>	<b>Remoto</b>	<b>Cluster</b>
X	X			

Isso controla se as mensagens do acionador são gravadas em uma fila de inicialização para fazer com que um aplicativo seja iniciado para atender a fila. Essa ação é uma das seguintes:

#### **TCOFF**

Mensagens do acionador não necessárias.

Nenhuma mensagem do acionador deve ser gravada para esta fila O valor *TriggerType* é irrelevante neste caso.

#### **TCON**

Mensagens do acionador necessárias..

As mensagens do acionador devem ser gravadas para essa fila, quando ocorrerem os eventos do acionador apropriados

Para determinar o valor desse atributo, use o seletor IATRGC com a chamada MQINQ.. Para alterar o valor desse atributo, use a chamada MQSET..

### **IBM i TriggerData (sequência de caracteres de 64 bytes) no IBM i**

Dados do acionador.

<i>Tabela 804. Tipos de fila aos quais este atributo se aplica</i>				
<b>Local</b>	<b>Modelo</b>	<b>Alias</b>	<b>Remoto</b>	<b>Cluster</b>
X	X			

Estes são dados de formato livre que o gerenciador de filas insere na mensagem do acionador quando uma mensagem que chega a essa fila faz com que uma mensagem do acionador seja gravada na fila de inicialização

O conteúdo desses dados não é significativo para o gerenciador da fila. Isso é significativo para o aplicativo acionador-monitor que processa a fila de iniciação ou para o aplicativo que é iniciado pelo monitor acionador

A cadeia de caracteres não pode conter nenhum valor nulo É preenchido à direita com espaços em branco, se necessário.

Para determinar o valor deste atributo, use o seletor CATRGD com a chamada MQINQ. Para alterar o valor desse atributo, use a chamada MQSET.. O comprimento deste atributo é fornecido por LNTRGD

### **IBM i TriggerDepth (número inteiro assinado de 10 dígitos) em IBM i**

Profundidade do acionador.

<i>Tabela 805. Tipos de fila aos quais este atributo se aplica</i>				
<b>Local</b>	<b>Modelo</b>	<b>Alias</b>	<b>Remoto</b>	<b>Cluster</b>
X	X			

Este é o número de mensagens de prioridade *TriggerMsgPriority* ou maior que devem estar na fila antes de uma mensagem do acionador ser gravada Isso se aplica quando o *TriggerType* é configurado como TTDPTH O valor de *TriggerDepth* é um ou maior. Caso contrário, esse atributo não será usado

Para determinar o valor desse atributo, use o seletor IATRGD com a chamada MQINQ.. Para alterar o valor desse atributo, use a chamada MQSET..

**IBM i TriggerMsgPrioridade (número inteiro assinado de 10 dígitos) em IBM i**

Prioridade de mensagem de limite para acionadores em IBM MQ for IBM i.

*Tabela 806. Tipos de fila aos quais este atributo se aplica*

Local	Modelo	Alias	Remoto	Cluster
X	X			

Essa é a prioridade da mensagem abaixo da qual as mensagens não contribuem para a geração de mensagens do acionador (ou seja, o gerenciador de filas ignora essas mensagens ao determinar se uma mensagem do acionador deve ser gerada.) *TriggerMsgPriority* pode estar no intervalo zero (mais baixo) a *MaxPriority* (mais alto; consulte [“Atributos para o gerenciador de filas no IBM i”](#) na página 1449); um valor zero faz com que todas as mensagens contribuam para a geração de mensagens do acionador.

Para determinar o valor desse atributo, utilize o seletor IATRGP com a chamada MQINQ. Para alterar o valor desse atributo, use a chamada MQSET..

**IBM i TriggerType (número inteiro assinado de 10 dígitos) em IBM i**

Tipo de acionador.

*Tabela 807. Tipos de fila aos quais este atributo se aplica*

Local	Modelo	Alias	Remoto	Cluster
X	X			

Isso controla as condições nas quais as mensagens do acionador são gravadas como resultado de mensagens que chegam nesta fila. O valor é um dos seguintes:

**TTNONE**

Nenhuma mensagem do acionador..

Nenhuma mensagem do acionador é gravada como um resultado de mensagens nesta fila Isso tem o mesmo efeito que configurar *TriggerControl* para TCOFF..

**TTFRST**

Mensagem do acionador quando a profundidade da fila vai de 0 a 1.

Uma mensagem do acionador é gravada sempre que o número de mensagens de prioridade *TriggerMsgPriority* ou maior na fila for alterado de 0 para 1.

**TTEVRY**

Mensagem do acionador para cada mensagem..

Uma mensagem do acionador é gravada sempre que uma mensagem de prioridade *TriggerMsgPriority* ou maior chega à fila

**TTDPTH**

Mensagem do acionador quando o limite de profundidade for excedido.

Uma mensagem do acionador é gravada sempre que o número de mensagens de prioridade *TriggerMsgPriority* ou maior na fila for igual a ou exceder *TriggerDepth* Após a mensagem do acionador ter sido gravada, *TriggerControl* será configurado como TCOFF para evitar acionamento adicional até que ele seja explicitamente ativado novamente



Para determinar o valor desse atributo, use o seletor IATRGT com a chamada MQINQ.. Para alterar o valor desse atributo, use a chamada MQSET..

### **IBM i** *Uso (número inteiro assinado de 10 dígitos) em IBM i*

Uso da fila..

<i>Tabela 808. Tipos de fila aos quais este atributo se aplica</i>				
Local	Modelo	Alias	Remoto	Cluster
X	X			

Isso indica para que a fila é usada.. O valor é um dos seguintes:

#### **USNORM**

Utilização normal.

Essa é uma fila que os aplicativos normais usam ao colocar e obter mensagens; a fila não é uma fila de transmissão

#### **USTRAN**

Fila de transmissão.

Esta é uma fila usada para reter mensagens destinadas a gerenciadores de filas remotas Quando um aplicativo normal envia uma mensagem para uma fila remota, o gerenciador de filas locais armazena a mensagem temporariamente na fila de transmissão apropriada em um formato especial.. Em seguida, um agente do canal de mensagens lê a mensagem da fila de transmissão e transporta a mensagem para o gerenciador de filas remotas. Para obter mais informações sobre filas de transmissão, consulte [Filas de transmissão](#).

Somente aplicativos privilegiados podem abrir uma fila de transmissão para OOOOUT para colocar mensagens diretamente nela.. Normalmente, espera-se que apenas aplicativos utilitários façam isso. Deve-se tomar cuidado para que o formato de dados da mensagem esteja correto (consulte "[MQXQH \(Transmission-queue header\) em IBM i](#)" na página 1290 ), caso contrário, poderão ocorrer erros durante o processo de transmissão... O contexto não é transmitido ou configurado a menos que uma das opções de contexto PM\* seja especificada.

Para determinar o valor desse atributo, use o seletor IAUSAG com a chamada MQINQ..

### **IBM i** *XmitQName (sequência de caractere de 48 bytes) no IBM i*

Nome da fila de transmissão.

<i>Tabela 809. Tipos de fila aos quais este atributo se aplica</i>				
Local	Modelo	Alias	Remoto	Cluster
			X	

Se esse atributo não estiver em branco quando uma abertura ocorrer, seja para uma fila remota ou para uma definição de alias do gerenciador de fila, ele especificará o nome da fila de transmissão local a ser usada para encaminhar a mensagem

Se *XmitQName* estiver em branco, a fila local com o mesmo nome que *RemoteQMGrName* será usada como a fila de transmissão Se não houver nenhuma fila com o nome *RemoteQMGrName*, a fila identificada pelo atributo do gerenciador de filas **DefXmitQName** será usada

Este atributo será ignorado se a definição estiver sendo usada como um alias do gerenciador de filas e *RemoteQMGrName* for o nome do gerenciador de fila local Ele também será ignorado se a definição for usada como uma definição de alias de fila de resposta.

Para determinar o valor desse atributo, use o seletor CAXQN com a chamada MQINQ. O comprimento desse atributo é fornecido pelo LNQNN

## Atributos para Listas de Nomes

Este tópico resume os atributos específicos para listas de nomes. Os atributos são descritos em ordem alfabética

**Nota:** Os nomes dos atributos mostrados são os nomes usados com as chamadas MQINQ e MQSET

### Descrições de atributos

Um objeto da lista de nomes possui os seguintes atributos:

#### **AlterationDate (sequência de caracteres de 12 bytes)**

Data da última alteração da definição.

Esta é a data em que a definição foi mudada pela última vez. O formato da data é YYYY-MM-DD, preenchido com dois espaços em branco finais para tornar o comprimento de 12 bytes.

Para determinar o valor desse atributo, use o seletor CAALTD com a chamada MQINQ.. O comprimento deste atributo é fornecido por LNDATE

#### **AlterationTime (sequência de caracteres de 8 bytes)**

Horário em que a definição foi mudada pela última vez

Este é o horário em que a definição foi alterada pela última vez. O formato do horário é HH.MM.SS

Para determinar o valor desse atributo, use o seletor CAALTT com a chamadas MQINQ. O comprimento desse atributo é fornecido por LNTIME

#### **NameCount (número inteiro assinado de 10 dígitos)**

O número de nomes na lista de nomes.

É maior ou igual a zero. O seguinte valor é definido:

##### **NCMXNL**

O número máximo de nomes em uma lista de nomes.

Para determinar o valor desse atributo, use o seletor IANAMC com a chamada MQINQ..

#### **NamelistDesc (sequência de caracteres de 64 bytes).**

Descrição da lista de nomes

Este é um campo que pode ser usado para comentário descritivo; seu valor é estabelecido pelo processo de definição.. O conteúdo do campo não é significativo para o gerenciador de filas, mas o gerenciador de filas pode requerer que o campo contenha apenas caracteres que possam ser exibidos.. Ele não pode conter nenhum caractere nulo; se necessário, será preenchido à direita com espaços em branco. Em uma instalação DBCS, este campo pode conter caracteres DBCS (sujeito a um comprimento máximo de campo de 64 bytes).

**Nota:** Se esse campo contiver caracteres que não estão no conjunto de caracteres do gerenciador de filas (conforme definido pelo atributo do gerenciador de filas **CodedCharSetId**), esses caracteres poderão ser convertidos incorretamente se esse campo for enviado para outro gerenciador de filas..

Para determinar o valor deste atributo, use o seletor CALSTD com a chamada MQINQ.

O comprimento desse atributo é fornecido por LNNLD

#### **NamelistName (sequência de caracteres de 48 bytes).**

Nome da lista de nomes..

Este é o nome de uma lista de nomes definida no gerenciador de fila local.

Cada lista de nomes tem um nome diferente dos nomes de outras listas de nomes pertencentes ao gerenciador de filas, mas pode duplicar os nomes de outros objetos do gerenciador de filas de diferentes tipos (por exemplo, filas).

Para determinar o valor desse atributo, use o seletor CALSTN com a chamada MQINQ..

O comprimento deste atributo é fornecido por LNNLN

### **Nomes (sequência de caracteres de 48 bytes x NameCount)**

Uma lista de nomes do *NameCount*

Cada nome é o nome de um objeto definido para o gerenciador de filas locais. Para obter mais informações sobre nomes de objetos, consulte [Nomeando IBM MQ objetos](#)

Para determinar o valor desse atributo, use o seletor CANAMS com a chamada MQINQ

O comprimento de cada nome na lista é fornecido por LNOBJN.



## **Atributos para definições de processo em IBM i**

Este tópico resume os atributos específicos para definições de processo. Os atributos são descritos em ordem alfabética

**Nota:** Os nomes dos atributos mostrados são os nomes usados com as chamadas MQINQ e MQSET. Quando comandos MQSC são usados para definir, alterar ou exibir atributos, nomes abreviados alternativos são usados; consulte [Comandos MQSC](#) para obter detalhes.

### **Descrições de atributos**

Um objeto de definição de processo possui os seguintes atributos:

#### **AlterationDate (sequência de caracteres de 12 bytes)**

Data da última alteração da definição.

Esta é a data em que a definição foi mudada pela última vez. O formato da data é YYYY-MM-DD, preenchido com dois espaços em branco finais para tornar o comprimento de 12 bytes.

Para determinar o valor desse atributo, use o seletor CAALTD com a chamada MQINQ.. O comprimento deste atributo é fornecido por LNDATE

#### **AlterationTime (sequência de caracteres de 8 bytes)**

Horário em que a definição foi mudada pela última vez

Este é o horário em que a definição foi alterada pela última vez. O formato do horário é HH.MM.SS

Para determinar o valor desse atributo, use o seletor CAALTT com a chamadas MQINQ. O comprimento desse atributo é fornecido por LNTIME

#### **ApplId (sequência de caracteres de 256 bytes)**

O identificador do aplicativo.

Esta é uma sequência de caracteres que identifica o aplicativo a ser iniciado.. Essas informações são para uso por um aplicativo de monitor acionador que processa mensagens na fila de inicialização; as informações são enviadas à fila de inicialização como parte da mensagem do acionador.

O significado de *ApplId* é determinado pelo aplicativo do monitor acionador. O monitor acionador fornecido pelo IBM MQ requer que *ApplId* seja o nome de um programa executável

A cadeia de caracteres não pode conter nenhum valor nulo. É preenchido à direita com espaços em branco, se necessário.

Para determinar o valor desse atributo, use o seletor CAAPPI com a chamada MQINQ.. O comprimento desse atributo é fornecido por LNPROA

## **ApplType (número inteiro assinado de 10 dígitos)**

Tipo de Aplicativo.

Isso identifica a natureza do programa a ser iniciado em resposta ao recebimento de uma mensagem do acionador. Essas informações são para uso por um aplicativo de monitor acionador que processa mensagens na fila de inicialização; as informações são enviadas à fila de inicialização como parte da mensagem do acionador.

*ApplType* pode ter qualquer valor. É possível usar os seguintes valores para tipos padrão; os tipos de aplicativos definidos pelo usuário são restritos a valores no intervalo de ATUFST a ATULST:

### **àsCICS**

Transação CICS .

### **AT400**

IBM i .

### **ATUFST**

Valor mais baixo para o tipo de aplicativo definido pelo usuário.

### **ATULST**

Valor mais alto para o tipo de aplicativo definido pelo usuário.

Para determinar o valor desse atributo, use o seletor IAAPPT com a chamada MQINQ..

## **EnvData (sequência de caracteres de 128 bytes).**

Dados do ambiente

Esta é uma sequência de caracteres que contém informações relacionadas ao ambiente pertencentes ao aplicativo a ser iniciado. Essas informações são para uso por um aplicativo de monitor acionador que processa mensagens na fila de inicialização; as informações são enviadas à fila de inicialização como parte da mensagem do acionador.

O significado de *EnvData* é determinado pelo aplicativo do monitor acionador. O monitor acionador fornecido por IBM MQ anexa *EnvData* à lista de parâmetros transmitida para o aplicativo iniciado. A lista de parâmetros consiste na estrutura MQTMC2 , seguida por um espaço em branco, seguido por *EnvData* com espaços em branco finais removidos.

A cadeia de caracteres não pode conter nenhum valor nulo. É preenchido à direita com espaços em branco, se necessário.

Para determinar o valor desse atributo, use o seletor CAENVD com a chamadas MQINQ. O comprimento desse atributo é fornecido por LNPROE

## **ProcessDesc (sequência de caracteres de 64 bytes)**

Descrição do processo..

Este é um campo que pode ser usado para comentário descritivo.. O conteúdo do campo não é significativo para o gerenciador de filas, mas o gerenciador de filas pode requerer que o campo contenha apenas caracteres que possam ser exibidos... Ele não pode conter nenhum caractere nulo; se necessário, será preenchido à direita com espaços em branco. Em uma instalação DBCS, o campo pode conter caracteres DBCS (sujeito a um comprimento máximo de campo de 64 bytes).

**Nota:** Se esse campo contiver caracteres que não estão no conjunto de caracteres do gerenciador de filas (conforme definido pelo atributo do gerenciador de filas **CodedCharSetId** ), esses caracteres poderão ser convertidos incorretamente se esse campo for enviado para outro gerenciador de filas..

Para determinar o valor desse atributo, use o seletor CAPROD com a chamada MQINQ..

O comprimento deste atributo é fornecido por LNPROD

## **ProcessName (sequência de caracteres de 48 bytes)**

Nome do processo.

Este é o nome de uma definição de processo definida no gerenciador de fila local.

Cada definição de processo tem um nome diferente dos nomes de outras definições de processo pertencentes ao gerenciador de fila. Mas o nome da definição de processo pode ser o mesmo que os nomes de outros objetos do gerenciador de filas de diferentes tipos (por exemplo, filas).

Para determinar o valor deste atributo, use o seletor CAPRON com a chamada MQINQ.

O comprimento deste atributo é fornecido por LNPRON

### **UserData (sequência de caracteres de 128 bytes),**

Dados do usuário.

Esta é uma sequência de caracteres que contém informações sobre o usuário pertencentes ao aplicativo a ser iniciado. Essas informações são para uso por um aplicativo de monitor acionador que processa mensagens na fila de iniciação ou o aplicativo que é iniciado pelo monitor acionador. As informações são enviadas à fila de inicialização como parte da mensagem do acionador.

O significado de *UserData* é determinado pelo aplicativo do monitor acionador. O monitor acionador fornecido por IBM MQ passa *UserData* para o aplicativo iniciado como parte da lista de parâmetros. A lista de parâmetros consiste na estrutura MQTMC2 (contendo *UserData*), seguida por um espaço em branco, seguido por *EnvData* com espaços em branco finais removidos.

A cadeia de caracteres não pode conter nenhum valor nulo. É preenchido à direita com espaços em branco, se necessário.

Para determinar o valor desse atributo, use o seletor CAUSRD com a chamada MQINQ. O comprimento deste atributo é fornecido por LNPROU.

## **IBM i Atributos para o gerenciador de filas no IBM i**

Um resumo dos atributos do gerenciador de filas

Alguns atributos do gerenciador de fila são fixos para implementações específicas, enquanto outros podem ser mudados usando o comando MQSC ALTER QMGR. Os atributos também podem ser exibidos usando o comando DISPLAY QMGR. A maioria dos atributos do gerenciador de filas podem ser consultados abrindo um objeto OTQM especial e usando a chamada MQINQ com o identificador retornado.

A tabela a seguir resume os atributos específicos do gerenciador de filas. Os atributos são descritos em ordem alfabética.

**Nota:** Os nomes dos atributos mostrados nesta seção são os nomes usados com as chamadas MQINQ e MQSET. Quando comandos MQSC são usados para definir, alterar ou exibir atributos, nomes abreviados alternativos são usados; consulte [Comandos MQSC](#) para obter mais informações.

<i>Tabela 810. Atributos do gerenciador de filas</i>	
<b>Atributo</b>	<b>Descrição</b>
<u>AlterationDate</u>	Data em que a definição foi alterada pela última vez
<u>AlterationTime</u>	Horário em que a definição foi alterada pela última vez
<u>AuthorityEvent</u>	Controla se eventos de autorização (Não Autorizado) são gerados
<u>BridgeEvent</u>	Controla se eventos de ponte do IMS são gerados
<u>ChannelAutoDef</u>	Controla se a definição de canal automático é permitida.
<u>ChannelAutoDefEvent</u>	Controla se os eventos de definição automática do canal são gerados
<u>ChannelAutoDefExit</u>	Nome da saída de usuário para definição de canal automática
<u>ChannelEvent</u>	Controla se os eventos do canal são gerados.
<u>ClusterCacheTipo</u>	Controla se o cache do cluster é fixo em tamanho ou dinamicamente dimensionado

<i>Tabela 810. Atributos do gerenciador de filas (continuação)</i>	
<b>Atributo</b>	<b>Descrição</b>
<u>ClusterWorkloadData</u>	Dados do usuário para saída de carga de trabalho do cluster
<u>ClusterWorkloadExit</u>	Nome da saída de usuário para gerenciamento de carga de trabalho do cluster
<u>ClusterWorkloadLength</u>	Comprimento máximo de dados da mensagem transmitidos à saída de carga de trabalho do cluster
<u>CodedCharSetId</u>	Identificador do conjunto de caracteres codificados
<u>CommandEvent</u>	Controla se as mensagens de evento de comando são enfileirados...
<u>CommandInputQName</u>	Nome da fila de entrada do comando
<u>CommandLevel</u>	Nível de comando
<u>ConfigurationEvent</u>	Evento de configuração
<u>DeadLetterQName</u>	Nome da fila de mensagens não entregues
<u>DefClusterXmitQueueTipo</u>	Tipo de fila de transmissão do cluster padrão
<u>DefXmitQName</u>	Nome da Fila de Transmissão Padrão
<u>DistLists</u>	Suporte à lista de distribuição
<u>InhibitEvent</u>	Controla se os eventos de inibição (Inhibit Get e Inhibit Put) são gerados
<u>LocalEvent</u>	Controla se eventos de erro locais são gerados
<u>LoggerEvent</u>	Controla se eventos do log de recuperação são gerados
<u>MaxHandles</u>	Número máximo de identificadores
<u>MaxMsgLength</u>	Tamanho máximo da mensagem em bytes
<u>MaxPriority</u>	Prioridade máxima
<u>MaxUncommittedMsgs</u>	Número máximo de mensagens não confirmadas em uma unidade de trabalho
<u>PerformanceEvent</u>	Controla se eventos relacionados ao desempenho são gerados
<u>Plataforma</u>	Plataforma na qual o gerenciador de filas está em execução
<u>PubSubMode</u>	Se o mecanismo de publicação / assinatura e a interface de publicação / assinatura enfileirada estão em execução
<u>QMgrDesc</u>	Descrição do gerenciador de filas
<u>QMgrIdentifier</u>	Identificador interno exclusivo do gerenciador de filas
<u>QMgrName</u>	Nome do gerenciador de filas
<u>RemoteEvent</u>	Controla se eventos de erro remotos são gerados
<u>RepositoryName</u>	Nome do cluster para o qual este gerenciador de filas fornece serviços de repositório
<u>RepositoryNamelist</u>	Nome do objeto da lista de nomes contendo nomes de clusters para os quais este gerenciador de filas fornece serviços do repositório..
<u>SSLCRLNamelist</u>	Nome do objeto da lista de nomes contendo nomes de objetos de informações sobre autenticação (consulte a Nota 1)

<i>Tabela 810. Atributos do gerenciador de filas (continuação)</i>	
<b>Atributo</b>	<b>Descrição</b>
<u>SSLEvent</u>	Controla se os eventos TLS são gerados
<u>SSLKeyRepository</u>	Localização do repositório de chaves TLS (consulte a Nota 1)
<u>SSLKeyResetCount</u>	Determina o número de bytes não criptografados enviados e recebidos em uma conversa TLS antes que a chave de criptografia seja renegociada
<u>StartStopEvent</u>	Controla se eventos de início e parada são gerados
<u>SyncPoint</u>	Disponibilidade do ponto de sincronização
<u>TraceRouteRecording</u>	Controla a gravação de informações de rota de rastreamento para mensagens
<u>TreeLifeTime</u>	A duração, em segundos, de tópicos não administrativos
<u>TriggerInterval</u>	Accionador-intervalo de mensagem
<b>Notas:</b>	
1. Este atributo não pode ser consultado usando a chamada MQINQ e não está descrito nesta seção Para obter mais informações sobre esse atributo, consulte <a href="#">Change Queue Manager</a> .	

### **IBM i** *AlterationDate (sequência de caracteres de 12 bytes) em IBM i*

Data da última alteração da definição.

Esta é a data em que a definição foi mudada pela última vez O formato da data é YYYY-MM-DD, preenchido com dois espaços em branco finais para tornar o comprimento de 12 bytes.

Para determinar o valor desse atributo, use o seletor CAALTD com a chamada MQINQ.. O comprimento deste atributo é fornecido por LNDATE

### **IBM i** *AlterationTime (sequência de caracteres de 8 bytes) no IBM i*

Horário em que a definição foi mudada pela última vez

Este é o horário em que a definição foi alterada pela última vez O formato do horário é HH.MM.SS

Para determinar o valor desse atributo, use o seletor CAALTT com a chamadas MQINQ. O comprimento desse atributo é fornecido por LNTIME

### **IBM i** *AuthorityEvent (número inteiro assinado de 10 dígitos) no IBM i*

Controla se eventos de autorização (Não Autorizado) são gerados.

O atributo AuthorityEvent deve ser definido para um dos seguintes valores:

#### **EVRDIS**

Relatório de eventos desativado.

#### **EVRENA**

Relatório de eventos ativado

Para obter mais informações sobre os eventos, consulte [Monitoramento de eventos](#)

Para determinar o valor desse atributo, use o seletor IAAUTE com a chamada MQINQ...

### **IBM i** *BridgeEvent (sequência de caracteres) no IBM i*

Esse atributo determina se as mensagens do evento de ponte do IMS são colocadas no SYSTEM.ADMIN.CHANNEL.EVENT fila. Ele é suportado apenas no z/OS

**ChannelAutoDef (número inteiro assinado de 10 dígitos) em IBM i**

Controla se a definição de canal automática é permitida

Este atributo controla a definição automática de canais do tipo CTCVR e CTSVCN. Observe que a definição automática de canais do CTCLSD está sempre ativada. Isso pode ter um dos seguintes valores:

**CHADDI**

Definição automática de canal desativada.

**CHADEN**

Definição automática de canal ativada.

Para determinar o valor deste atributo, use o seletor IACAD com a chamada MQINQ.

**ChannelAutoDefEvent (número inteiro assinado de 10 dígitos) no IBM i**

Controla se eventos de definição automática de canal são gerados.

Isto aplica-se a canais do tipo CTCVR, CTSVCN e CTCLSD. Isso pode ter um dos seguintes valores:

**EVRDIS**

Relatório de eventos desativado.

**EVRENA**

Relatório de eventos ativado

Para obter mais informações sobre eventos, consulte [Monitoramento e desempenho](#).

Para determinar o valor deste atributo, use o seletor IACADE com a chamada MQINQ.

**ChannelAutoDefExit (sequência de caracteres de 20 bytes) no IBM i**

Nome da saída de usuário para definição de canal automática.

Se esse nome não estiver em branco e *ChannelAutoDef* tiver o valor CHADEN, a saída será chamada toda vez que o gerenciador de filas estiver prestes a criar uma definição de canal. Isto aplica-se a canais do tipo CTCVR, CTSVCN e CTCLSD. A saída pode, então, executar um dos seguintes procedimentos:

- Permita a criação da definição de canal para continuar sem mudança
- Modifique os atributos da definição de canal criada.
- Suprimir a criação do canal inteiramente.

Para determinar o valor desse atributo, use o seletor CACADX com a chamadas MQINQ. O comprimento deste atributo é fornecido por LNEXN

**ChannelEvent (sequência de caracteres) no IBM i.**

Determina se as mensagens de eventos do canal são geradas.

Este atributo determina se as mensagens de evento do canal são colocadas no SYSTEM.ADMIN.CHANNEL.EVENT e, em caso afirmativo, quais tipos de mensagens serão enfileiradas (por exemplo, 'channel started', 'channel stopped', 'channel not activated'). Antes da implementação desse atributo, a única maneira de evitar que mensagens de eventos do canal fossem enfileiradas era excluir a fila de destino.

Esse atributo também permite coletar somente eventos de ponte do IMS (porque agora é possível desativar eventos do canal, eles não são colocados na mesma fila). O mesmo se aplica aos eventos TLS que também podem ser coletados sem precisar coletar eventos do canal também.

Esse atributo também permite coletar somente eventos significativos (por exemplo, quando os canais têm erros, não quando eles iniciam e param normalmente).

O valor para o atributo ChannelEvent pode ser um dos seguintes:

- EVREXP (apenas os seguintes eventos do canal são gerados: RC2279, RC2283, RC2284, RC2295, RC2296).



- EVRENA (todos os eventos do canal são gerados; ou seja, além dos eventos gerados pelo EVREXP, os eventos RC2282, e RC2283 também são gerados)
- EVRDIS (nenhum evento de canal é gerado; este é o valor padrão inicial do gerenciador de filas)...

Para determinar o valor desse atributo, use o seletor IACHNE com a chamada MQINQ..

### **IBM i ClusterCacheTipo (sequência de caractere de 32 bytes) no IBM i**

Controla se o cache do cluster é de tamanho fixo ou de tamanho dinâmico.

Essa é uma sequência de caracteres de 32 bytes definida pelo usuário que é transmitida para a saída de carga de trabalho do cluster quando ela é chamada. Se não houver dados a serem transmitidos para a saída, a sequência estará em branco.

Para determinar o valor desse atributo, use o seletor CACLWD com a chamada MQINQ..

### **IBM i ClusterWorkloadDados (sequência de caracteres de 32 bytes) em IBM i.**

Dados do usuário para a saída de carga de trabalho do cluster.

Essa é uma sequência de caracteres de 32 bytes definida pelo usuário que é transmitida para a saída de carga de trabalho do cluster quando ela é chamada. Se não houver dados a serem transmitidos para a saída, a sequência estará em branco.

Para determinar o valor desse atributo, use o seletor CACLWD com a chamada MQINQ..

### **IBM i ClusterWorkloadSaída (sequência de caracteres de 20 bytes) no IBM i**

Nome da saída de usuário para o gerenciamento de cargas de trabalho do cluster.

Se esse nome não estiver em branco, a saída será chamada toda vez que uma mensagem for colocada em uma fila de clusters ou movida de uma fila do emissor de clusters para outra. A saída pode então aceitar a instância da fila selecionada pelo gerenciador de filas como o destino para a mensagem ou selecionar outra instância da fila.

Para determinar o valor desse atributo, use o seletor CACLWX com a chamada MQINQ.. O comprimento deste atributo é fornecido por LNEXN.

### **IBM i ClusterWorkloadComprimento (número inteiro assinado de 10 dígitos) em IBM i**

Comprimento máximo de dados da mensagem transmitidos para a saída de carga de trabalho do cluster.

Este é o comprimento máximo de dados da mensagem que é transmitido para a saída de carga de trabalho do cluster. O comprimento real dos dados passados para a saída é o mínimo dos seguintes:

- O comprimento da mensagem.
- O atributo **MaxMsgLength** do Gerenciador de Filas.
- O atributo **ClusterWorkloadLength** ..

Para determinar o valor desse atributo, use o seletor IACLWL com a chamada MQINQ..

### **IBM i CodedCharSetId (número inteiro assinado de 10 dígitos) em IBM i**

Identificador do conjunto de caracteres codificados.

Isso define o conjunto de caracteres usado pelo gerenciador de filas para todos os campos de sequência de caracteres definidos no MQI, como os nomes dos objetos e a data e hora de criação da fila. O conjunto de caracteres deve ser um que tenha caracteres de byte único para os caracteres válidos em nomes de objetos. Ele não se aplica aos dados do aplicativo transportados na mensagem. O valor depende do ambiente:

- No IBM i, o valor é aquele configurado no ambiente quando o gerenciador de filas é criado pela primeira vez.

Para determinar o valor desse atributo, use o seletor IACCSI com a chamada MQINQ..

## **IBM i** *CommandEvent (número inteiro) no IBM i*

Controla se as mensagens são colocadas em uma fila local quando os comandos são emitidos

Isso controla se as mensagens são gravadas em uma nova fila de eventos, SYSTEM.ADMIN.COMMAND.EVENT sempre que comandos forem emitidos. Esse recurso é útil para notificação de rastreamento de comando e para diagnóstico de problemas Para consultar sobre o atributo do gerenciador de filas CommandEvent , use o novo seletor de atributo iacev com um dos seguintes valores:

- EVRENA-as mensagens do evento de comando são geradas e colocadas na fila para todos os comandos bem-sucedidos
- EVND-mensagens de eventos de comando são geradas e colocadas na fila para todos os comandos bem-sucedidos diferentes do comando DISPLAY (MQSC) e do comando Inquire (PCF).
- EVRDIS-mensagens de evento de comando não são geradas ou colocadas na fila (este é o valor padrão inicial do gerenciador de filas).

Para determinar o valor desse atributo, use o seletor CMDEV com a chamada MQINQ.

## **IBM i** *CommandInputQName (sequência de caractere de 48 bytes) no IBM i*

Nome da fila de entrada de comandos

CommandInputQName é o nome da fila de entrada de comandos definida no gerenciador de fila local.. É uma fila para a qual os usuários podem enviar comandos, se autorizados a fazê-lo. O nome da fila depende do ambiente:

- No IBM i, o nome da fila é SYSTEM.ADMIN.COMMAND.QUEUE e apenas comandos PCF podem ser enviados para ele.. No entanto, um comando MQSC pode ser enviado para essa fila se o comando MQSC estiver incluído em um comando PCF do tipo CMESC Para obter mais informações sobre o comando Escape, consulte [Escape](#)..

Para determinar o valor desse atributo, use o seletor CACMDQ com a chamada MQINQ.. O comprimento desse atributo é fornecido pelo LNQNN

## **IBM i** *CommandLevel (número inteiro assinado de 10 dígitos) no IBM i*

Nível de Comando Isso indica o nível de comandos de controle do sistema suportados pelo gerenciador de filas

O nível é um dos seguintes valores:

### **CML800**

O nível 800 de comandos de controle do sistema.

Esse valor é retornado pelos seguintes aplicativos:

- IBM MQ for IBM i
  - Versão 8.0

### **CML900**

O nível 900 dos comandos de controle do sistema

Esse valor é retornado pelos seguintes aplicativos:

- IBM MQ for IBM i
  - Versão 9.0

### **CML910**

Nível 910 de comandos de controle do sistema.

Esse valor é retornado pelos seguintes aplicativos:

- IBM MQ for IBM i
  - Versão 9.1

## CML920

Nível 920 de comandos de controle do sistema.

Esse valor é retornado pelos seguintes aplicativos:

- IBM MQ for IBM i
  - Versão 9.2

## CML930

Nível 930 de comandos de controle do sistema.

Esse valor é retornado pelos seguintes aplicativos:

- IBM MQ for IBM i
  - Versão 9.3

O conjunto de comandos de controle do sistema que corresponde a um determinado valor do atributo **CommandLevel** varia de acordo com o valor do atributo **Platform**; ambos devem ser usados para decidir quais comandos de controle do sistema são suportados.

Para determinar o valor desse atributo, use o seletor IACMDL com a chamada MQINQ.

## **ConfigurationEvent em IBM i**

Controla se os eventos de configuração são gerados e enviados para o SYSTEM.ADMIN.CONFIG.EVENT do objeto padrão da fila.

O atributo ConfigurationEvent pode ser um dos seguintes valores:

- EVRENA
- EVRDIS

Se o atributo ConfigurationEvent for configurado como EVRENA, e determinados comandos forem emitidos com êxito por runmqsc ou PCF, os eventos de configuração serão gerados e enviados para o SYSTEM.ADMIN.CONFIG.EVENT fila. Eventos para os seguintes comandos são emitidos, mesmo se um comando alter não alterar o objeto envolvido. Os comandos para os quais os eventos de configuração são gerados e enviados são:

- DEFINE/ALTER AUTHINFO
- DEFINE/ALTER CHANNEL
- DEFINE/ALTER NAMELIST
- DEFINE/ALTER PROCESS
- DEFINE/ALTER QLOCAL (a menos que seja uma fila dinâmica temporária)
- DEFINE/ALTER QMODEL/QALIAS/QREMOTE
- DELETE AUTHINFO
- DELETE CHANNEL
- DELETE NAMELIST
- DELETE PROCESS
- DELETE QLOCAL (a menos que seja uma fila dinâmica temporária)
- DELETE QMODEL/QALIAS/QREMOTE
- ALTER QMGR (a menos que o atributo CONFIGEV esteja desativado e não seja alterado para ativado)
- REFRESH QMGR
- Uma chamada MQSET, diferente de uma fila dinâmica temporária.

Os eventos não são gerados (se ativados) nas seguintes circunstâncias:

- O comando ou chamada MQSET falha.

- O gerenciador de filas não pode colocar a mensagem do evento sobre a fila de eventos O comando ainda deve ser concluído com êxito
- Filas dinâmicas temporárias
- Mudanças de atributos internos feitas diretamente ou implicitamente (não por MQSET ou comando); isso afeta TRIGGER, CURDEPTH, ipprocs e opprocs, Qdphiev, Qdploev e QSVCI EV.
- Quando a fila de eventos de configuração for mudada, embora uma mensagem de evento seja gerada para essa mudança quando uma Atualização for solicitada
- Armazenando em cluster mudanças pelos comandos REFRESH/RESET CLUSTER e RESUME/SUSPEND QMGR.
- Criando ou excluindo um gerenciador de filas.

## **DeadLetterQName (sequência de caractere de 48 bytes) em IBM i**

Nome da fila de mensagens não entregues.

Este é o nome de uma fila definida no Gerenciador de Filas Local As mensagens serão enviadas para esta fila se não puderem ser roteados para o seu destino correto

Por exemplo, as mensagens são colocadas nesta fila quando:

- Uma mensagem chega a um gerenciador de filas, destinado para uma fila que ainda não está definida nesse gerenciador de fila
- Uma mensagem chega em um gerenciador de filas, mas a fila para a qual ela está destinada não pode recebê-la porque, possivelmente:
  - A fila está cheia
  - As solicitações put são inibidas
  - O nó de envio não tem autoridade para colocar mensagens na fila

Aplicativos também podem colocar mensagens na fila de mensagens não entregues.

As mensagens de relatório são tratadas da mesma maneira que as mensagens ordinárias; se a mensagem de relatório não puder ser entregue em sua fila de destino (geralmente a fila especificada pelo campo *MDRQ* no descritor de mensagens da mensagem original), a mensagem de relatório será colocada na fila de mensagens não entregues.

**Nota:** Mensagens que passaram seu tempo de expiração (consulte o campo *MDEXP* descrito em [“MQMD \(Descritor de mensagens\) em IBM i”](#) na página 1148 ) **não** são transferidos para esta fila quando eles são descartadas No entanto, uma mensagem de relatório de expiração (ROEXP) ainda é gerada e enviada para a fila *MDRQ* , se solicitado pelo aplicativo de envio.

As mensagens não são colocadas na fila de mensagens não entregues (mensagens não entregues) quando o aplicativo que emitiu a solicitação put foi notificado de forma síncrona do problema com o código de razão retornado pela chamada MQPUT ou MQPUT1 (por exemplo, uma mensagem colocada em uma fila local para a qual as solicitações put são inibidas).

As mensagens na fila de mensagens não entregues (não entregues) às vezes têm seus dados da mensagem do aplicativo prefixados com uma estrutura MQDLH. Essa estrutura contém informações extras que indicam por que a mensagem foi colocada na fila de mensagens não entregues (mensagens não entregues). Consulte [“MQDLH \(cabecalho de devoluções\) em IBM i”](#) na página 1101 para obter mais detalhes sobre essa estrutura

Esta fila deve ser uma fila local, com um atributo **Usage** de USNORM

Se uma fila de mensagens não entregues não for suportada por um gerenciador de filas ou uma não tiver sido definida, o nome será todo em branco. Todos os gerenciadores de filas do IBM MQ suportam uma fila de mensagens não entregues, mas, por padrão, ela não está definida.

Se a fila de mensagens não entregues (mensagens não entregues) não estiver definida ou estiver cheia ou inutilizável por alguma outra razão, uma mensagem que teria sido transferida para ela por um agente do canal de mensagens será retida na fila de transmissão.

Para determinar o valor deste atributo, utilize o seletor CADLQ com a chamada MQINQ. O comprimento desse atributo é fornecido pelo LNQNN

### ***DefClusterXmitQueueTipo (número inteiro assinado de 10 dígitos)***

O atributo DefClusterXmitQueueTipo controla qual fila de transmissão é selecionada por padrão pelos canais do emissor de clusters para obter mensagens, para enviar as mensagens para os canais do receptor de clusters.

Os valores de **DefClusterXmitQueueType** são MQCLXQ\_SCTQ ou MQCLXQ\_CHANNEL.

#### **MQCLXQ\_SCTQ**

Todos os canais do emissor de clusters enviam mensagens a partir do SYSTEM.CLUSTER.TRANSMIT.QUEUE. O correlID de mensagens colocadas na fila de transmissão identifica para qual canal do emissor de clusters a mensagem é destinada.

SCTQ é configurado quando um gerenciador de filas é definido

#### **MQCLXQ\_CHANNEL**

Cada canal do emissor de clusters envia mensagens a partir de uma fila de transmissão diferente. Cada fila de transmissão é criada como uma fila dinâmica permanente a partir da fila de modelos SYSTEM.CLUSTER.TRANSMIT.MODEL.QUEUE.

Se o atributo do gerenciador de filas, DefClusterXmitQueueType, for configurado como CHANNEL A configuração padrão será alterada para os canais do emissor de clusters que estão sendo associados às filas de transmissão de cluster individuais. As filas de transmissão são filas permanentes e dinâmicas criadas a partir da fila modelo SYSTEM.CLUSTER.TRANSMIT.MODEL.QUEUE. Cada fila de transmissão é associada a um canal do emissor de clusters. Conforme um canal do emissor de clusters atende uma fila de transmissão do cluster, a fila de transmissão contém mensagens apenas para um gerenciador de filas em um cluster. É possível configurar clusters para que cada gerenciador de filas em um cluster contenha apenas uma fila de clusters. Neste caso, o tráfego de mensagens de um gerenciador de filas para cada fila de clusters é transferido separadamente de mensagens para outras filas.

Para consultar o valor, chame MQINQ ou envie um comando PCF Inquire Queue Manager (MQCMD\_INQUIRE\_Q\_MGR), configurando o seletor MQIA\_DEF\_CLUSTER\_XMIT\_Q\_TYPE. Para alterar o valor, envie um comando PCF Change Queue Manager (MQCMD\_CHANGE\_Q\_MGR), configurando o seletor MQIA\_DEF\_CLUSTER\_XMIT\_Q\_TYPE.

#### **Referências relacionadas**

[Alterar Gerenciador de Filas](#)

[Consultar Gerenciador de Filas](#)

[“MQINQ \(Consultar sobre atributos do objeto\) em IBM i” na página 1352](#)

A chamada MQINQ retorna uma matriz de números inteiros e um conjunto de cadeias de caracteres contendo os atributos de um objeto.

### **IBM i *DefXmitQName (sequência de caractere de 48 bytes) em IBM i***

Nome da fila de transmissão padrão..

Este é o nome da fila de transmissão que é usada para a transmissão de mensagens para gerenciadores de filas remotas, se não houver nenhuma outra indicação de qual fila de transmissão usar

Se não houver uma fila de transmissão padrão, o nome estará inteiramente em branco O valor inicial deste atributo está em branco

Para determinar o valor desse atributo, use o seletor CADXQN com a chamada MQINQ.. O comprimento desse atributo é fornecido pelo LNQNN

### **IBM i *DistLists (número inteiro assinado de 10 dígitos) em IBM i***

Suporte à lista de distribuições

Indica se o gerenciador de filas locais suporta listas de distribuição nas chamadas MQPUT e MQPUT1. Isso pode ter um dos seguintes valores:

## **DLSUPP**

Listas de distribuição suportadas.

## **DLNSUP**

Listas de distribuição não suportadas.

Para determinar o valor desse atributo, use o seletor IADIST com a chamada MQINQ.

## **IBM i InhibitEvent (número inteiro assinado de 10 dígitos) no IBM i**

Controla se eventos de inibição (Inhibit Get e Inhibit Put) são gerados.

Isso pode ter um dos seguintes valores:

### **EVRDIS**

Relatório de eventos desativado.

### **EVRENA**

Relatório de eventos ativado

Para obter mais informações sobre eventos, consulte [Monitoramento e desempenho](#)..

Para determinar o valor desse atributo, use o seletor IAINHE com a chamada MQINQ.

## **IBM i LocalEvent (número inteiro assinado de 10 dígitos) no IBM i**

Controla se os eventos de erro locais são gerados

O valor é um dos seguintes:

### **EVRDIS**

Relatório de eventos desativado.

### **EVRENA**

Relatório de eventos ativado

Para obter mais informações sobre eventos, consulte [Monitoramento de eventos](#)

Para determinar o valor desse atributo, use o seletor IALCLE com a chamada MQINQ..

## **IBM i LoggerEvent (número inteiro assinado de 10 dígitos) em IBM i**

Controla se os eventos do criador de logs de recuperação são gerados

Isso pode ter um dos seguintes valores:

### **ATIVADO**

Os eventos do criador de logs são gerados

### **DISABLED**

Os eventos do criador de logs não são gerados Este é o valor padrão inicial dos gerenciadores de filas.

Para obter mais informações sobre eventos, consulte [Monitoramento e desempenho](#)..

## **IBM i MaxHandles (número inteiro assinado de 10 dígitos) no IBM i**

O número máximo de identificadores

Este é o número máximo de identificadores abertos que qualquer tarefa pode utilizar simultaneamente Cada chamada MQOPEN bem-sucedida para uma única fila (ou para um objeto que não seja uma fila) usa uma manipulação de um identificador. Essa manipulação torna-se disponível para reutilização quando o objeto é encerrado No entanto, quando uma lista de distribuição é aberta, cada fila na lista de distribuição é alocada um identificador separado e para que a chamada MQOPEN use tantos identificadores quanto houver filas na lista de distribuição. Isso deve ser considerado ao decidir sobre um valor adequado para *MaxHandles*.

A chamada MQPUT1 executa uma chamada MQOPEN como parte de seu processamento; como resultado, MQPUT1 usa tantas manipulações como MQOPEN, mas as manipulações são usadas apenas para a duração da própria chamada MQPUT1

O valor está no intervalo de 1 a 999 999 999. No IBM i, o valor padrão é 256.

Para determinar o valor desse atributo, use o seletor IAMHND com a chamada MQINQ..

**IBM i** **MaxMsgComprimento (número inteiro assinado de 10 dígitos) em IBM i**  
Comprimento máximo da mensagem, em bytes.

Esse é o comprimento da mensagem *física* mais longa que pode ser manipulado pelo gerenciador de filas. No entanto, como o atributo do gerenciador de fila **MaxMsgLength** pode ser configurado independentemente do atributo da fila **MaxMsgLength**, a mensagem física mais longa que pode ser colocada em uma fila é o menor desses dois valores.

Se o gerenciador de filas suportar a segmentação, será possível para um aplicativo colocar uma mensagem *lógica* que seja maior que o menor dos dois atributos **MaxMsgLength**, mas somente se o aplicativo especificar o sinalizador MFSEGA no MQMD. Se esse sinalizador for especificado, o limite superior para o comprimento de uma mensagem lógica será 999 999 999 bytes, mas, geralmente, as restrições de recursos impostas pelo sistema operacional ou pelo ambiente no qual o aplicativo está em execução resultarão em um limite inferior.

O limite inferior para o atributo **MaxMsgLength** é 32 KB (32 768 bytes). No IBM i, o comprimento máximo da mensagem é 100 MB (104 857 600 bytes).

Para determinar o valor deste atributo, use o seletor IAMLEN com a chamada MQINQ.

**IBM i** **MaxPriority (número inteiro assinado de 10 dígitos) no IBM i**  
Prioridade máxima..

Esta é a prioridade máxima da mensagem suportada pelo gerenciador de fila. As prioridades variam de zero (mais baixo) a *MaxPriority* (mais alto).

Para determinar o valor desse atributo, use o seletor IAMPRI com a chamada MQINQ.

**IBM i** **MaxUncommittedMsgs (número inteiro assinado de 10 dígitos) no IBM i**  
Número máximo de mensagens não consolidadas em uma unidade de trabalho.

Este é o número máximo de mensagens não consolidadas que podem existir em uma unidade de trabalho. O número de mensagens não confirmadas é a soma do seguinte, desde o início da unidade de trabalho atual:

- Mensagens colocadas pelo aplicativo com a opção PMSYP
- Mensagens recuperadas pelo aplicativo com a opção GMSYP
- Mensagens do acionador e mensagens do relatório COA geradas pelo gerenciador de filas para mensagens colocadas com a opção PMSYP
- Mensagens de relatório COD geradas pelo gerenciador de filas para mensagens recuperadas com a opção GMSYP

As seguintes mensagens não são contadas como não confirmadas:

- Mensagens colocadas ou recuperadas pelo aplicativo fora de uma unidade de trabalho
- Mensagens do acionador ou mensagens de relatório COA/COD geradas pelo gerenciador de filas como resultado de mensagens colocadas ou recuperadas fora de uma unidade de trabalho.
- Mensagens de relatório de expiração geradas pelo gerenciador de filas (mesmo que a chamada que está causando a mensagem de relatório de expiração tenha especificado GMSYP)
- Mensagens de evento geradas pelo gerenciador de filas (mesmo se a chamada estiver causando a mensagem de evento especificada PMSYP ou GMSYP)

**Nota:**

1. As mensagens de relatório de exceção são geradas pelo Message Channel Agent (MCA) ou pelo aplicativo e, portanto, são tratadas da mesma maneira que as mensagens ordinárias colocadas ou recuperadas pelo aplicativo.



2. Quando uma mensagem ou segmento é colocado com a opção PMSYP, o número de mensagens não confirmadas é incrementado em um, independentemente de quantas mensagens físicas realmente resultam da colocação. (Mais de uma mensagem física pode resultar se o gerenciador de filas precisar subdividir a mensagem ou segmento.)
3. Quando uma lista de distribuição é colocada com a opção PMSYP, o número de mensagens não confirmadas é incrementado em um *para cada mensagem física gerada*. Isso pode ser tão pequeno quanto um ou tão grande quanto o número de destinos na lista de distribuição.

O limite inferior desse atributo é 1; o limite superior é 999 999 999.

Para determinar o valor desse atributo, use o seletor IAMUNC com a chamada MQINQ..

### **PerformanceEvent (número inteiro assinado de 10 dígitos) no IBM i**

Controla se os eventos relacionados ao desempenho são gerados

PerformanceEvent pode ter um dos seguintes valores:

#### **EVRDIS**

Relatório de eventos desativado.

#### **EVRENA**

Relatório de eventos ativado

Para obter mais informações sobre os eventos, consulte [Monitoramento de eventos](#)

Para determinar o valor desse atributo, use o seletor IAPFME com a chamada MQINQ..

### **Plataforma (número inteiro assinado de 10 dígitos) no IBM i**

A plataforma na qual o gerenciador de filas está em execução

Indica o sistema operacional no qual o gerenciador de fila está em execução. O valor é:

#### **PL400**

IBM i.

### **Modo PubSub(número inteiro assinado de 10 dígitos) no IBM i**

Se o mecanismo de publicação / assinatura e a interface de publicação / assinatura enfileirada estão em execução, permitindo, portanto, que os aplicativos publiquem / assinem usando a interface de programação de aplicativos e as filas que estão sendo monitorados pela interface de publicação / assinatura enfileirada

Isso pode ter um dos seguintes valores:

#### **PSMCP**

O mecanismo de publicação/assinatura está em execução. Portanto, é possível publicar / assinar usando a interface de programação de aplicativos. A interface de publicação / assinatura enfileirada não está em execução, portanto, nenhuma mensagem que é colocada nas filas que são monitoradas pela interface de publicação / assinatura enfileirada é executada. Essa configuração é usada para compatibilidade com WebSphere Message Broker V6 ou versões anteriores usando esse gerenciador de filas, porque ele deve ler as mesmas filas a partir das quais a interface de publicação / assinatura enfileirada normalmente lê.

#### **PSMDS**

O mecanismo de publicação/assinatura e a interface de publicação/assinatura enfileirada não estão em execução. Portanto, não é possível publicar / assinar usando a interface de programação de aplicativos.. Quaisquer mensagens de publicação / assinatura que são colocadas nas filas que são monitoradas pela interface de publicação / assinatura enfileirada não têm ação.

#### **PSMEN**

O mecanismo de publicação/assinatura e a interface de publicação/assinatura enfileirada estão em execução. Portanto, é possível publicar / assinar usando a interface de programação de aplicativos e as filas que estão sendo monitoradas pela interface de publicação / assinatura enfileiradas Este é o valor padrão inicial do gerenciador de filas.



Para determinar o valor desse atributo, use o seletor PSMODE com a chamada MQINQ..

### **IBM i** **QMgrDesc (sequência de caracteres de 64 bytes) no IBM i** descrição do gerenciador de filas.

Este é um campo que pode ser usado para comentário descritivo.. O conteúdo do campo não é significativo para o gerenciador de filas, mas o gerenciador de filas pode requerer que o campo contenha apenas caracteres que possam ser exibidos... Ele não pode conter nenhum caractere nulo; se necessário, será preenchido à direita com espaços em branco. Em uma instalação DBCS, este campo pode conter caracteres DBCS (sujeito a um comprimento máximo de campo de 64 bytes).

**Nota:** Se esse campo contiver caracteres que não estão no conjunto de caracteres do gerenciador de filas (conforme definido pelo atributo do gerenciador de filas **CodedCharSetId**), esses caracteres poderão ser convertidos incorretamente se esse campo for enviado para outro gerenciador de filas..

No IBM i, o valor padrão é espaços em branco.

Para determinar o valor desse atributo, use o seletor CAQMD com a chamada MQINQ.. O comprimento deste atributo é fornecido pelo LNQMD

### **IBM i** **QMgrIdentifier (sequência de caractere de 48 bytes) em IBM i** Identificador interno exclusivo do gerenciador de filas.

Este é um nome exclusivo gerado internamente para o gerenciador de filas

Para determinar o valor desse atributo, use o seletor CAQMID com a chamadas MQINQ. O comprimento desse atributo é fornecido por LNQMID.

### **IBM i** **QMgrName (sequência de caractere de 48 bytes) no IBM i** Nome do gerenciador de filas.

Este é o nome do gerenciador de fila local, ou seja o nome do gerenciador de fila ao qual o aplicativo está conectado

Os primeiros 12 caracteres do nome são usados para construir um identificador de mensagem exclusivo (consulte o campo *MDMID* descrito em “[MQMD \(Descritor de mensagens\) em IBM i](#)” na página 1148 ) Portanto, os gerenciadores de filas que podem intercomunicar devem ter nomes que diferem nos primeiros 12 caracteres, para que os identificadores de mensagens sejam exclusivos na rede do gerenciador de filas.

Para determinar o valor desse atributo, use o seletor CAQMN com a chamada MQINQ.. O comprimento desse atributo é fornecido por LNQMN

### **IBM i** **RemoteEvent (número inteiro assinado de 10 dígitos) no IBM i**

Controla se os eventos de erro remotos são gerados

O valor é um dos seguintes:

#### **EVRDIS**

Relatório de eventos desativado.

#### **EVRENA**

Relatório de eventos ativado

Para obter mais informações sobre os eventos, consulte [Monitoramento de eventos](#)

Para determinar o valor deste atributo, use o seletor IARMTE com a chamada MQINQ.

### **IBM i** **RepositoryName (sequência de caractere de 48 bytes) em IBM i**

O nome do cluster para o qual esse gerenciador de filas fornece serviços de repositório

Este é o nome de um cluster para o qual este gerenciador de filas fornece um serviço do gerenciador do repositório Se o gerenciador de filas fornecer esse serviço para mais de um cluster, *RepositoryNameList* especificará o nome de um objeto de lista de nomes que identifica os clusters

e *RepositoryName* estará em branco. Pelo menos um de *RepositoryName* e *RepositoryNameList* deve estar em branco

Para determinar o valor deste atributo, use o seletor CARPN com a chamada MQINQ. O comprimento desse atributo é fornecido por LNQMNM

**IBM i**

### ***RepositoryNameList (sequência de caractere de 48 bytes) em IBM i***

Nome do objeto da lista de nomes que contém nomes de clusters para os quais este gerenciador de filas fornece serviços de repositório

Este é o nome de um objeto de lista de nomes que contém os nomes de clusters para os qual este gerenciador de fila fornece um serviço de gerenciador de repositório Se o gerenciador de filas fornecer esse serviço para apenas um cluster, o objeto da lista de nomes conterá apenas um nome Como alternativa, *RepositoryName* pode ser usado para especificar o nome do cluster, em cujo caso, *RepositoryNameList* está em branco Pelo menos um de *RepositoryName* e *RepositoryNameList* deve estar em branco

Para determinar o valor deste atributo, use o seletor CARPNL com a chamada MQINQ. O comprimento deste atributo é fornecido por LNNLN

**IBM i**

### ***SSLEvent (sequência de caracteres) no IBM i***

Determina se eventos TLS são gerados.

O valor é um dos seguintes:

- EVRENA (MQINQ/PCF/config event) ENABLED (MQSC): eventos TLS são gerados (ou seja, o evento RC2371 é gerado).
- EVRDIS (MQINQ/PCF/config event) DISABLED (MQSC): eventos TLS não são gerados. Este é o valor padrão inicial do gerenciador de filas.

Para determinar o valor desse atributo, use o seletor IASSLE com a chamada MQINQ.

**IBM i**

### ***SSLKeyResetContagem (número inteiro) em IBM i***

Determina o número total de bytes não criptografados que são enviados e recebidos dentro de uma conversa TLS, antes que a chave secreta seja renegociada O número de bytes inclui informações de controle enviadas pelo MCA (Message Channel Agent).

Esse valor é usado apenas por MCAs do canal TLS que iniciam a comunicação a partir desse gerenciador de filas (ou seja, o MCA do canal emissor em um emparelhamento de canais emissor e receptor).

Se o valor desse atributo for maior que 0 e as pulsações do canal estiverem ativadas para um canal, a chave secreta também será renegociada antes que os dados sejam enviados ou recebidos após uma pulsação do canal. A contagem de bytes até a próxima renegociação de chave secreta ser reconfigurada após cada renegociação bem-sucedida ocorrer.

O valor pode estar no intervalo de 0 a 999 999 999. Um valor 0 para esse atributo indica que a chave secreta nunca é renegociada. Se você especificar uma contagem de reconfiguração de chave secreta TLS no intervalo de 1 byte a 32 KB, os canais TLS usarão uma contagem de reconfiguração de chave secreta de 32 KB. Isso é para evitar o custo de processamento de reconfigurações de chave excessivas que ocorreriam para pequenos valores de reconfiguração de chave secreta TLS..

Quando o servidor SSL é um gerenciador de filas do IBM MQ e a reconfiguração de chave secreta e as pulsações de canal estão ativadas, a renegociação ocorre imediatamente após cada pulsação de canal...

Para determinar o valor desse atributo, use o seletor IASSRC com a chamada MQINQ..

**IBM i**

### ***StartStopEvento (número inteiro assinado de 10 dígitos) no IBM i***

Controla se eventos de início e parada são gerados.

Esse atributo pode ter um dos seguintes valores:

## **EVRDIS**

Relatório de eventos desativado.

## **EVRENA**

Relatório de eventos ativado

Para obter mais informações sobre os eventos, consulte [Monitoramento de eventos](#)

Para determinar o valor desse atributo, use o seletor IASSE com a chamada MQINQ.

## **IBM i SyncPoint (número inteiro assinado de 10 dígitos) no IBM i**

Disponibilidade do Ponto de Sincronização

Isso indica se o gerenciador de filas locais suporta unidades de trabalho e sincronização com as chamadas MQGET, MQPUT e MQPUT1 .

## **SPAVL**

Unidades de trabalho e ponto de sincronização disponíveis

## **SPNAVL**

Unidades de trabalho e sincronização não disponíveis.

Para determinar o valor desse atributo, use o seletor IASYNC com a chamada MQINQ..

## **IBM i TraceRouteGravação (número inteiro assinado de 10 dígitos) no IBM i**

Isso controla se as informações sobre mensagens são registradas à medida que fluem por um gerenciador de filas.

O valor é um dos seguintes:

- RECD: nenhuma anexação para rastrear mensagens de rota é permitida
- RECDQ: mensagens são colocadas em uma fila nomeada fixa
- RECDM: determinar usando a mensagem (esta é a configuração padrão inicial)

Para evitar que a mensagem de rota de rastreo permaneça no sistema, configure um valor de expiração maior que zero e especifique a opção de relatório RODISC. Para evitar que mensagens de relatório ou resposta permaneçam no sistema, configure a opção de relatório ROPDAE. Para obter mais informações, consulte [“Opções de relatório e sinalizações de mensagem no IBM i”](#) na página 1484.

Para determinar o valor deste atributo, use o seletor IATRGI com a chamada MQINQ.

## **IBM i TreeLifeTempo (número inteiro assinado de 10 dígitos) em IBM i**

O tempo de vida, em segundos, de tópicos não administrativos

Tópicos não administrativos são aqueles criados quando um aplicativo publica ou assina como uma sequência de tópicos que não existe como um nó administrativo. Quando esse nó não administrativo não possui mais nenhuma assinatura ativa, este parâmetro determina quanto tempo o gerenciador de filas aguardará antes de remover esse nó. Apenas tópicos não administrativos que estão em uso por uma assinatura durável permanecem após a reciclagem do gerenciador de filas.

Especifique um valor no intervalo de 0 a 604 000. Um valor 0 significa que tópicos não administrativos não são removidos pelo gerenciador de filas. O valor padrão inicial do gerenciador de filas é 1800.

Para determinar o valor desse atributo, use o seletor IATRLFT com a chamada MQINQ..

## **IBM i TriggerInterval (número inteiro assinado de 10 dígitos) no IBM i**

Intervalo da mensagem do acionador.

Esse é um intervalo de tempo (em milissegundos) usado para restringir o número de mensagens do acionador.. Isso é relevante apenas quando o *TriggerType* é TFRST Nesse caso, as mensagens do acionador normalmente são geradas apenas quando uma mensagem adequada chega na fila e a fila estava anteriormente vazia Sob determinadas circunstâncias, no entanto, uma mensagem do acionador

adicional pode ser gerada com o acionador TTFRST, mesmo se a fila não estava vazia. Essas mensagens do acionador adicionais não são geradas com mais frequência do que a cada *TriggerInterval* milissegundos

Para obter mais informações sobre acionamento, consulte [Canais de acionamento](#).

O valor está no intervalo de zero a 999 999 999. O valor padrão é 999 999 999.

Para determinar o valor deste atributo, use o seletor IATRGI com a chamada MQINQ.

## Aplicativos

Estas informações descrevem os programas de amostra entregues com o IBM MQ for IBM i para RPG. Além disso, saiba como construir aplicativos executáveis a partir dos programas que você escreve.

### Construindo seu Aplicativo

As publicações IBM i descrevem como construir aplicativos executáveis a partir dos programas gravados. Este tópico descreve as tarefas adicionais e as alterações nas tarefas padrão que devem ser executadas ao construir aplicativos IBM MQ for IBM i para execução em IBM i.

Além de codificar as chamadas MQI em seu código fonte, deve-se incluir as instruções de linguagem apropriadas para incluir os arquivos de cópias IBM MQ for IBM i para a linguagem RPG. Você deve se familiarizar com o conteúdo desses arquivos; seus nomes e uma breve descrição de seu conteúdo são fornecidos no texto a seguir.

#### **IBM MQ copiar arquivos em IBM i**

IBM MQ for IBM i fornece arquivos de cópia para ajudá-lo a gravar seus aplicativos na linguagem de programação RPG. Eles são adequados para uso com o WebSphere Development toolset (5722 WDS) ILE RPG 4 Compiler.

Os arquivos de cópia que o IBM MQ for IBM i fornece para ajudar na gravação de saídas de canal são descritos em [Programas de saída de canal para canais de sistema de mensagens](#).

Os nomes dos arquivos de cópia IBM MQ for IBM i para RPG têm o prefixo CMQ. Eles têm um sufixo de G ou H. Há arquivos de cópia separados que contêm as constantes nomeadas e um arquivo para cada uma das estruturas. Os arquivos de cópia são listados em [“Considerações sobre o idioma” na página 1044](#)

**Nota:** Para ILE RPG/400, eles são fornecidos como membros do arquivo QRPGLSRC na biblioteca QMQM..

As declarações de estrutura não contêm instruções DS .. Isso permite que o aplicativo declare uma estrutura de dados (ou uma estrutura de dados de várias ocorrências) codificando a instrução DS e usando a instrução /COPY para copiar no restante da declaração:

Para ILE RPG/400 , a instrução é:

```
D*..1.....2.....3.....4.....5.....6.....7
D* Declare an MQMD data structure
D MQMD          DS
D/COPY CMQMDG
```

### Preparando seus programas para execução

Para criar um aplicativo executável IBM MQ for IBM i , é necessário compilar o código-fonte gravado.

Para fazer isso para ILE RPG/400, é possível usar os comandos típicos IBM i , CRTRPGMOD e CRTPGM.

Após criar seu \*MODULE, é necessário especificar BNDSRVPGM (QMQM/LIBMQM) no comando CRTPGM. Isso inclui os vários procedimentos do IBM MQ em seu programa

Certifique-se de que a biblioteca que contém os arquivos de cópia (QMQM) esteja na lista de bibliotecas quando você executar a compilação

Para obter informações adicionais sobre considerações de programação, incluindo modos clientes, consulte [“Considerações sobre o idioma” na página 1044](#).

### **Interfaces para o gerenciador de ponto de sincronização externa do IBM i**

O IBM MQ for IBM i usa o controle de confirmação IBM i nativo como um coordenador de ponto de sincronização externo

Consulte o *IBM i Guia de Programação: Backup e Recuperação* para obter mais informações sobre os recursos de controle de confirmação do IBM i...

Para iniciar os recursos de controle de compromisso do IBM i, use o comando do sistema STRCMTCTL. Para finalizar o controle de confirmações, use o comando do sistema ENDCMTCTL.

**Nota:** O valor padrão de *Commitment definition scope* é \*ACTGRP. Isso deve ser definido como \*JOB para IBM MQ para IBM i. Por exemplo:

```
STRCMTCTL LCKLVL(*ALL) CMTSCOPE(*JOB)
```

Se você chamar MQPUT, MQPUT1 ou MQGET, especificando PMSYP ou GMSYP, após iniciar o controle de confirmação, IBM MQ for IBM i se inclui como um recurso de confirmação da API para a definição de confirmação. Essa é geralmente a primeira chamada em uma tarefa. Enquanto houver quaisquer recursos de confirmação da API registrados sob uma determinada definição de confirmação, você não pode finalizar o controle de compromisso para essa definição.

O IBM MQ for IBM i remove seu registro como um recurso de confirmação de API quando você se desconecta do gerenciador de fila, desde que não haja operações MQI pendentes na unidade de trabalho atual

Se você se desconectar do gerenciador de filas enquanto houver operações MQPUT, MQPUT1 ou MQGET pendentes na unidade de trabalho atual, o IBM MQ for IBM i permanece registrado como um recurso de confirmação de API para que ele seja notificado sobre a próxima confirmação ou retrocesso. Quando o próximo ponto de sincronização é atingido, o IBM MQ confirma ou retrocede as mudanças conforme necessário. É possível que um aplicativo se desconecte e reconecte a um gerenciador de filas durante uma unidade de trabalho ativa e execute operações MQGET e MQPUT adicionais dentro da mesma unidade de trabalho (essa é uma desconexão pendente).

Se você tentar emitir um comando do sistema ENDCMTCTL para essa definição de confirmação, a mensagem CPF8355 será emitida, indicando que as mudanças pendentes estavam ativas. Essa mensagem também aparece no log da tarefa quando a tarefa é finalizada. Para evitar isso, assegure-se de confirmar ou retroceder todas as operações pendentes do IBM MQ e de se desconectar do gerenciador de fila. Assim, o uso dos comandos COMMIT ou ROLLBACK antes de ENDCMTCTL deve permitir que o controle de confirmação final seja concluído com êxito.

Quando o controle de confirmação do IBM i é usado como um coordenador de ponto de sincronização externo, as chamadas MQCMIT, MQBACK e MQBEGIN podem não ser emitidas. Chamadas para essas funções falham com o código de razão RC2012.

Para confirmar ou retroceder (ou seja, para restaurar) sua unidade de trabalho, use uma das linguagens de programação que suporta o controle de confirmações. Por exemplo:

- Comandos CL: COMMIT e ROLLBACK
- Funções de programação ILE C: \_Rcommit e \_Rrollback
- RPG/400: COMMIT e ROLBK
- COBOL/400: COMMIT e ROLLBACK

### **Pontos de sincronização em CICS para os aplicativos IBM i**

IBM MQ for IBM i participa de unidades de trabalho com CICS. É possível usar o MQI em um aplicativo CICS para colocar e obter mensagens dentro da unidade de trabalho atual.

É possível usar o comando EXEC CICS SYNCPOINT para estabelecer um ponto de sincronização que inclui as operações do IBM MQ for IBM i. Para recuperar todas as mudanças até o ponto de sincronização anterior, é possível usar o comando EXEC CICS SYNCPOINT ROLLBACK.

Se você usar MQPUT, MQPUT1 ou MQGET com a opção PMSYP ou GMSYP configurada em um aplicativo CICS, não será possível efetuar logoff CICS até que IBM MQ for IBM i tenha removido seu registro como um recurso de confirmação da API. Portanto, você deve confirmar ou voltar quaisquer operações put ou get pendentes antes de desconectar do gerenciador de filas.. Isso permitirá que você efetue logoff de CICS

## Programas de amostra no IBM i ..

Este tópico descreve os programas de amostra entregues com o IBM MQ for IBM i para RPG. As amostras demonstram usos típicos do Message Queue Interface (MQI).

As amostras não são destinadas a demonstrar técnicas gerais de programação, portanto, algumas verificações de erro que você pode querer incluir em um programa de produção foram omitidas. No entanto, essas amostras são adequadas para uso como uma base para seus próprios programas de fila de mensagens.

O código de origem para todas as amostras é fornecido com o produto; esta origem inclui comentários que explicam as técnicas de enfileiramento de mensagens demonstradas nos programas.

Há um conjunto de programas de amostra ILE:

### 1. Programas usando chamadas de protótipo para o MQI (chamadas de limite estático)

A origem existe em QMQMSAMP/QRPGLESRC. Os membros são denominados AMQ3xxx4, em que xxx indica a função de amostra. Os membros de cópia existem em QMQM/QRPGLESRC. Cada nome do membro possui um sufixo de G ou H

Tabela 811 na página 1466 fornece uma lista completa dos programas de amostra entregues com IBM MQ for IBM i e mostra os nomes dos programas em cada uma das linguagens de programação suportadas. Observe que todos os seus nomes começam com o prefixo AMQ, o quarto caractere no nome indica a linguagem de programação.

	<b>RPG (ILE)</b>
Amostras Put	AMQ3PUT4
Procurar amostras	AMQ3GBR4
Amostras Get	AMQ3GET4
Amostras de solicitação	AMQ3REQ4
Amostras de eco	AMQ3ECH4
Consultar amostras	AMQ3INQ4
Configurar amostras	AMQ3SET4
Amostra do Monitor do Acionador.	AMQ3TRG4
Amostra do Servidor Acionador	AMQ3SRV4

Além disso, a opção de amostra IBM MQ for IBM i inclui um arquivo de dados de amostra, AMQSDATA, que pode ser usado como entrada para determinados programas de amostra e programas CL de amostra que demonstram tarefas de administração.. As amostras de CL são descritas em [Administrando IBM i](#). Você poderia usar o programa CL de amostra para criar filas a serem usadas com os programas de amostra descritos neste tópico

Para obter informações sobre como executar os programas de amostra, consulte [“Preparando e executando os programas de amostra no IBM i”](#) na página 1467

### **Recursos demonstrados nos programas de amostra no IBM i**

Uma tabela que mostra as técnicas demonstradas pelos programas de amostra do IBM MQ for IBM i

Algumas técnicas ocorrem em mais de um programa de amostra, mas apenas um programa é listado na tabela. Todas as filas abertas e fechadas de amostras usando as chamadas MQOPEN e MQCLOSE, portanto, essas técnicas não são listadas separadamente na tabela.

<i>Tabela 812. Programas de amostra que demonstram o uso do MQI</i>	
<b>Técnica</b>	<b>RPG (ILE)</b>
Usando as chamadas MQCONN e MQDISC	AMQ3ECH4 ou AMQ3INQ4
Conectando e desconectando implicitamente	AMQ3PUT4
Colocando mensagens usando a chamada MQPUT	AMQ3PUT4
Colocando uma única mensagem usando a chamada MQPUT1	AMQ3ECH4 ou AMQ3INQ4
Respondendo a uma mensagem de solicitação	AMQ3INQ4
Obtendo mensagens (nenhuma espera)	AMQ3GBR4
Obtendo mensagens (aguardar com um limite de tempo)	AMQ3GET4
Obtendo mensagens (com conversão de dados)	AMQ3ECH4
Procurando uma fila	AMQ3GBR4
Usando uma fila de entrada compartilhada	AMQ3INQ4
Usando uma fila de entrada exclusiva	AMQ3REQ4
Usando a chamada MQINQ	AMQ3INQ4
Usando a chamada MQSET	AMQ3SET4
Usando uma fila de resposta	AMQ3REQ4
Solicitando mensagens de exceção	AMQ3REQ4
Aceitando uma mensagem truncada	AMQ3GBR4
Usando um nome de fila resolvido	AMQ3GBR4
Processamento do acionador	AMQ3SRV4 ou AMQ3TRG4

**Nota:** Todos os programas de amostra produzem um arquivo de spool que contém os resultados do processamento

### **Preparando e executando os programas de amostra no IBM i**

Antes de poder executar os programas de amostra IBM MQ for IBM i, você deve compilá-los como faria com qualquer outro aplicativo IBM MQ for IBM i. Para fazer isso, é possível usar os comandos CRTPGMOD e CRTPGM do IBM i

Ao criar os programas AMQ3xxx4, você deve especificar BNDSRVPGM (QMQM/LIBMQM) no comando CRTPGM. Isso inclui os vários procedimentos IBM MQ em seu programa.

Os programas de amostra são fornecidos na biblioteca QMQMSAMP como membros de QRPGLSRC Eles usam os arquivos de cópia fornecidos na biblioteca QMQM, portanto, certifique-se de que essa biblioteca



esteja na lista de bibliotecas ao compilá-los. O compilador RPG fornece mensagens de informações porque as amostras não usam muitas das variáveis declaradas nos arquivos de cópia.

## Executando os programas de amostra

É possível usar suas filas ao executar as amostras ou compilar e executar AMQSAMP4 para criar algumas filas de amostra. A origem para este programa é enviada no arquivo QCLSRC na biblioteca QMQMSAMP. Ele pode ser compilado usando o comando CRTCLPGM.

Para chamar um dos programas de amostra, use um comando como:

```
CALL PGM(QMQMSAMP/AMQ3PUT4) PARM('Queue_Name','Queue_Manager_Name')
```

Em que Queue\_Name e Queue\_Manager\_Name devem ter 48 caracteres de comprimento, que você consegue preenchendo Queue\_Name e Queue\_Manager\_Name com o número necessário de espaços em branco.

Para os programas de amostra Inquire e Set, as definições de amostra criadas pelo AMQSAMP4 causam o acionamento das versões C dessas amostras. Se você desejar acionar as versões de RPG, deverá alterar as definições de processo SYSTEM.SAMPLE.ECHOPROCESS e SYSTEM.SAMPLE.INQPROCESS e SYSTEM.SAMPLE.SETPROCESS. É possível usar o comando CHGMQMPCRC (descrito em [Alterar MQ Processo \(CHGMQMPCRC\)](#)) para fazer isso ou editar e executar AMQSAMP4 com a definição alternativa.

### ***O programa de amostra Put em IBM i***

O programa de amostra Put, AMQ3PUT4, coloca as mensagens em uma fila usando a chamada MQPUT

Para iniciar o programa, chame o programa e forneça o nome de sua fila de destino como um parâmetro de programa. O programa coloca um conjunto de mensagens fixas na fila; essas mensagens são obtidas do bloco de dados no final do código fonte do programa. Um programa put de amostra é AMQ3PUT4 na biblioteca QMQMSAMP..

Utilizando este programa de exemplo, o comando é:

```
CALL PGM(QMQMSAMP/AMQ3PUT4) PARM('Queue_Name','Queue_Manager_Name')
```

Em que Queue\_Name e Queue\_Manager\_Name devem ter 48 caracteres de comprimento, que você consegue preenchendo Queue\_Name e Queue\_Manager\_Name com o número necessário de espaços em branco.

### **Design do programa de amostra Put**

O programa usa a chamada MQOPEN com a opção OOOUT para abrir a fila de destino para colocar mensagens. Os resultados são de saída para um arquivo em spool. Se ele não puder abrir a fila, o programa gravará uma mensagem de erro contendo o código de razão retornado pela chamada MQOPEN. Para manter o programa simples, nesta e em chamadas MQI subsequentes, o programa usa valores padrão para muitas das opções.

Para cada linha de dados contidos no código fonte, o programa lê o texto em um buffer e usa a chamada MQPUT para criar uma mensagem de datagrama contendo o texto dessa linha. O programa continua até que ele atinja o final da entrada ou a chamada MQPUT falhará. Se o programa atingir o final da entrada, ele fechará a fila usando a chamada MQCLOSE.

### ***O programa de amostra Browse em IBM i***

O programa de amostra Browse, AMQ3GBR4, procura mensagens em uma fila usando a chamada MQGET.

O programa recupera cópias de todas as mensagens na fila especificada quando você chama o programa; as mensagens permanecem na fila. Você poderia usar o SYSTEM.SAMPLE.LOCAL; execute o programa de amostra Put primeiro para colocar algumas mensagens na fila.. Você poderia usar a fila SYSTEM.SAMPLE.ALIAS, que é um nome de alias para a mesma fila local. O programa continua até atingir o final da fila ou uma chamada MQI falhar.



Um exemplo de um comando para chamar o programa RPG é:

```
CALL PGM(QMQMSAMP/AMQ3GBR4) PARM('Queue_Name','Queue_Manager_Name')
```

Em que `Queue_Name` e `Queue_Manager_Name` devem ter 48 caracteres de comprimento, que você consegue preenchendo `Queue_Name` e `Queue_Manager_Name` com o número necessário de espaços em branco. Portanto, se estiver usando `SYSTEM.SAMPLE.LOCAL` como sua fila de destino, serão necessários 29 caracteres em branco.

## Design do programa de amostra Browse

O programa abre a fila de destino usando a chamada `MQOPEN` com a opção `OBRW`. Se não for possível abrir a fila, o programa grava uma mensagem de erro em seu arquivo de spool, contendo o código de razão retornado pela chamada `MQOPEN`.

Para cada mensagem na fila, o programa usará a chamada `MQGET` para copiar a mensagem da fila, em seguida, exibirá os dados contidos na mensagem. A chamada `MQGET` usa estas opções:

### GMBRWN

Após a chamada `MQOPEN`, o cursor de navegação é posicionado logicamente antes da primeira mensagem na fila, portanto, essa opção faz com que a *primeira* mensagem seja retornada quando a primeira chamada for feita.

### GMNWT

O programa não espera se não houver mensagens na fila.

### GMATM

A chamada `MQGET` especifica um buffer de tamanho fixo. Se uma mensagem for maior do que esse buffer, o programa exibe a mensagem truncada, juntamente com um aviso de que a mensagem foi truncada.

O programa demonstra como você deve limpar os campos `MDMID` e `MDCID` da estrutura `MQMD` após cada chamada `MQGET` porque a chamada configura esses campos para os valores contidos na mensagem que ele recupera. Desmarcar esses campos significa que sucessivas chamadas `MQGET` recuperam as mensagens na ordem em que elas são retidas na fila.

O programa continua até o final da fila; aqui, a chamada `MQGET` retorna o código de razão `RC2033` (nenhuma mensagem disponível) e o programa exibe uma mensagem de aviso. Se a chamada `MQGET` falhar, o programa gravará uma mensagem de erro que contém o código de razão em seu arquivo em spool.

O programa, então, fecha a fila usando a chamada `MQCLOSE`.

## O programa de amostra Get em IBM i

O programa `Get sample`, `AMQ3GET4`, obtém mensagens de uma fila usando a chamada `MQGET`.

Quando o programa é chamado, ele remove mensagens da fila especificada. Você poderia usar o `SYSTEM.SAMPLE.LOCAL`; execute o programa de amostra `Put` primeiro para colocar algumas mensagens na fila.. Você poderia utilizar o `SYSTEM.SAMPLE.ALIAS`, que é um nome alternativo para a mesma fila local. O programa continua até a fila estar vazia ou uma chamada `MQI` falhar.

Um exemplo de um comando para chamar o programa RPG é:

```
CALL PGM(QMQMSAMP/AMQ3GET4) PARM('Queue_Name','Queue_Manager_Name')
```

em que `Queue_Name` e `Queue_Manager_Name` devem ter 48 caracteres de comprimento, que você consegue preenchendo `Queue_Name` e `Queue_Manager_Name` com o número necessário de espaços em branco. Portanto, se estiver usando `SYSTEM.SAMPLE.LOCAL` como sua fila de destino, serão necessários 29 caracteres em branco.

## Design do programa de amostra Get

O programa abre a fila de destino para obter mensagens; ele usa a chamada MQOPEN com a opção OOINPQ.. Se não for possível abrir a fila, o programa gravará uma mensagem de erro contendo o código de razão retornado pela chamada MQOPEN em seu arquivo de spool

Para cada mensagem na fila, o programa usa a chamada MQGET para remover a mensagem da fila e, em seguida, exibe os dados contidos na mensagem. A chamada MQGET usa a opção GMWT, especificando um intervalo de espera (*GMWI*) de 15 segundos, para que o programa aguarde esse período se não houver mensagens na fila. Se nenhuma mensagem chegar antes de esse intervalo expirar, a chamada falhará e retornará o código de razão RC2033 (nenhuma mensagem disponível).

O programa demonstra como você deve limpar os campos *MDMID* e *MDCID* da estrutura MQMD após cada chamada MQGET porque a chamada configura esses campos para os valores contidos na mensagem que ele recupera. Desmarcar esses campos significa que sucessivas chamadas MQGET recuperam as mensagens na ordem em que elas são retidas na fila.

A chamada MQGET especifica um buffer de tamanho fixo. Se uma mensagem for maior do que esse buffer, a chamada falhará e o programa irá parar.

O programa continua até a chamada MQGET retornar o código de razão RC2033 (nenhuma mensagem disponível) ou a chamada MQGET falhar. Se a chamada falhar, o programa exibirá uma mensagem de erro que contém o código de razão.

O programa, então, fecha a fila usando a chamada MQCLOSE.

## O programa de amostra Request em IBM i

O programa de amostra Request, AMQ3REQ4, demonstra o processamento do cliente / servidor. A amostra é o cliente que coloca mensagens de solicitação em uma fila que é processada por um programa servidor. Ele aguarda o programa do servidor colocar uma mensagem de resposta em uma fila de resposta.

A amostra Request coloca uma série de mensagens de solicitação em uma fila usando a chamada MQPUT. Essas mensagens especificam SYSTEM.SAMPLE.REPLY como a fila de resposta. O programa aguarda mensagens de resposta e, em seguida, as exibe. As respostas serão enviadas somente se a fila de destino (que chamaremos de *fila do servidor*) está sendo processado por um aplicativo do servidor ou se um aplicativo for acionado para esse propósito (os programas de amostra Inquire e Set foram projetados para serem acionados.) A amostra espera 5 minutos para que a primeira resposta chegue (para permitir tempo para um aplicativo do servidor ser acionado) e 15 segundos para respostas subsequentes, mas ela pode terminar sem obter nenhuma resposta

Para iniciar o programa, chame o programa e forneça o nome de sua fila de destino como um parâmetro de programa. O programa coloca um conjunto de mensagens fixas na fila; essas mensagens são obtidas do bloco de dados no final do código fonte do programa.

## Design do programa de amostra Request

O programa abre a fila do servidor para que possa colocar mensagens. Ele usa a chamada MQOPEN com a opção OOOUT. Se não puder abrir a fila, o programa exibirá uma mensagem de erro contendo o código de razão retornado pela chamada MQOPEN.

O programa abre, então, a fila de resposta chamada SYSTEM.SAMPLE.REPLY para que possa receber mensagens de resposta. Para isso, o programa usa a chamada MQOPEN com a opção OOINPX. Se não puder abrir a fila, o programa exibirá uma mensagem de erro contendo o código de razão retornado pela chamada MQOPEN.

Para cada linha de entrada, o programa então lê o texto em um buffer e usa a chamada MQPUT para criar uma mensagem de solicitação que contém o texto dessa linha. Nesta chamada, o programa usa a opção de relatório ROEXCD para solicitar que qualquer mensagem de relatório enviada sobre a mensagem de pedido inclua os primeiros 100 bytes dos dados da mensagem.. O programa continua até que ele atinja o final da entrada ou a chamada MQPUT falhará.

O programa usa, então, a chamada MQGET para remover mensagens de resposta da fila e exibe os dados contidos nas respostas. A chamada MQGET usa a opção GMWT, especificando um intervalo de espera (*GMWT*) de 5 minutos para a primeira resposta (para permitir tempo para um aplicativo do servidor ser acionado) e 15 segundos para respostas subsequentes. O programa espera esses períodos se não houver nenhuma mensagem na fila. Se nenhuma mensagem chegar antes de esse intervalo expirar, a chamada falhará e retornará o código de razão RC2033 (nenhuma mensagem disponível). A chamada também utiliza a opção GMATM, de modo que as mensagens maiores que o tamanho do buffer declarado são truncadas.

O programa demonstra como você deve limpar os campos *MDMID* e *MDCOD* da estrutura MQMD após cada chamada MQGET porque a chamada configura esses campos para os valores contidos na mensagem que ele recupera. Desmarcar esses campos significa que sucessivas chamadas MQGET recuperam as mensagens na ordem em que elas são retidas na fila.

O programa continua até a chamada MQGET retornar o código de razão RC2033 (nenhuma mensagem disponível) ou a chamada MQGET falhar. Se a chamada falhar, o programa exibirá uma mensagem de erro que contém o código de razão.

O programa então fecha a fila do servidor e a fila de resposta usando a chamada MQCLOSE. [Tabela 813 na página 1471](#) mostra as mudanças no programa de amostra Echo necessárias para executar os programas de amostra Inquire e Set.

**Nota:** Os detalhes para o programa de amostra Echo estão incluídos como uma referência.

<i>Tabela 813. Detalhes do programa de amostra Cliente / Servidor</i>		
<b>Nome do programa</b>	<b>Fila SYSTEM/SAMPLE</b>	<b>Programa iniciado</b>
echo	echo	AMQ3ECH4
Pedir informações	INQ	AMQ3INQ4
Configurar	SET	AMQ3SET4

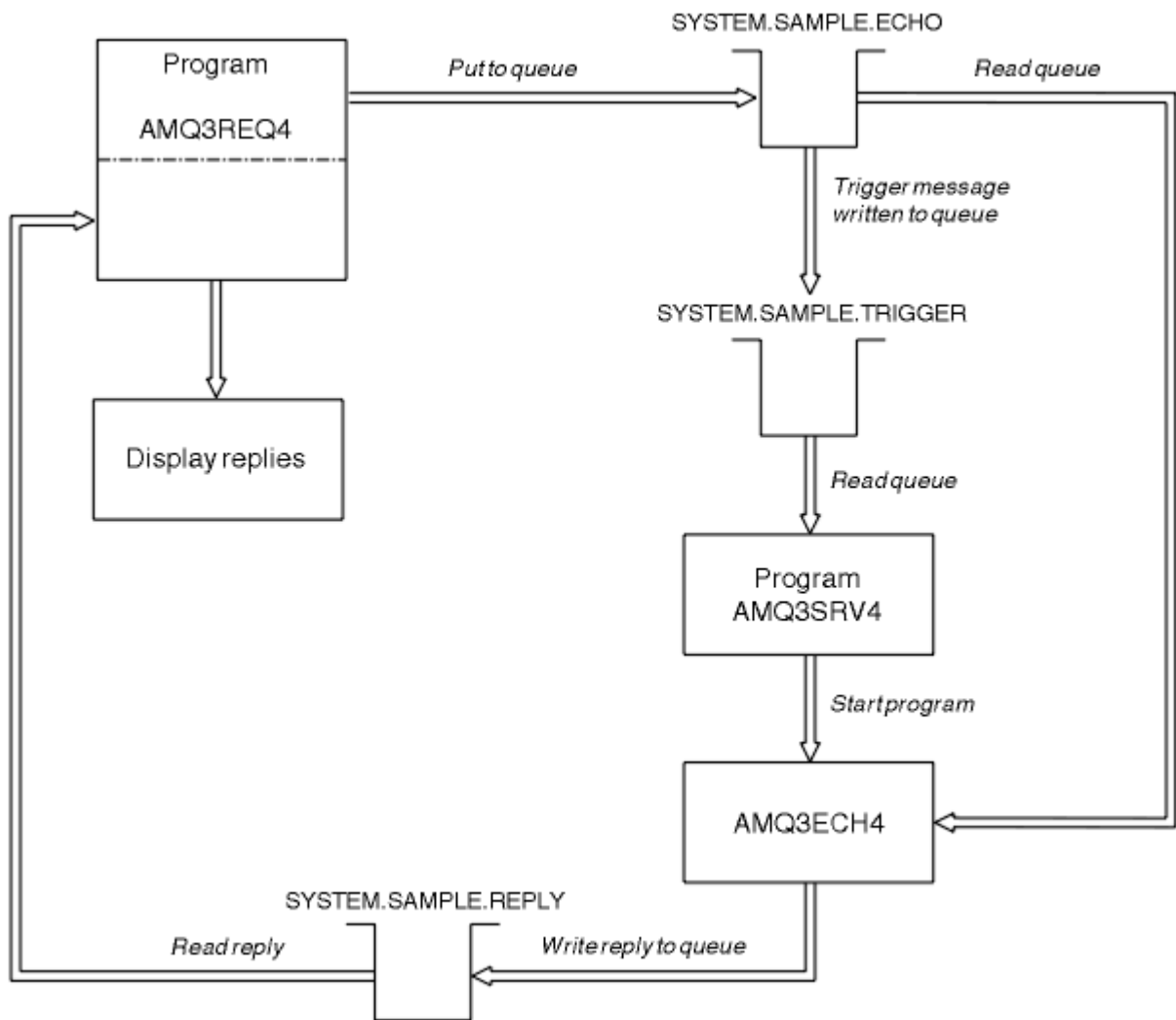


Figura 9. Fluxograma do programa Cliente / Servidor de Amostra (Eco)

**IBM i** Usando o acionamento com a amostra Request no IBM i

Para executar a amostra usando o acionamento, inicie o programa do servidor acionador, AMQ3SRV4, com relação à fila de inicialização necessária em uma tarefa, em seguida, inicie AMQ3REQ4 em outra tarefa.

Isso significa que o servidor do acionador está pronto quando o programa de amostra Request envia uma mensagem.

**Nota:**

1. As amostras usam a fila SYSTEM SAMPLE TRIGGER como a fila de inicialização para o SYSTEM.SAMPLE.ECHO, SYSTEM.SAMPLE.INQ, ou filas locais SYSTEM.SAMPLE.SET Como alternativa, é possível definir sua fila de inicialização.
2. As definições de amostra criadas por AMQSAMP4 fazem com que a versão C da amostra seja acionada Se desejar acionar a versão de RPG, deve-se alterar as definições de processo SYSTEM.SAMPLE.ECHOPROCESS e SYSTEM.SAMPLE.INQPROCESS e SYSTEM.SAMPLE.SETPROCESS É possível usar o comando CHGMQMPRC (consulte [Mudar MQ Processo \(CHGMQMPRC\)](#) para obter mais detalhes) para fazer isso ou editar e executar sua própria versão de AMQSAMP4.
3. Você deve compilar o programa do servidor acionador a partir da origem fornecida em QMQMSAMP/QRPGLESRC.

Dependendo do processo acionador que você deseja executar, AMQ3REQ4 deve ser chamado com o parâmetro que especifica mensagens de solicitação a serem colocadas em uma dessas filas do servidor de amostra:

- SYSTEM.SAMPLE.ECHO (para os programas de amostra Echo)
- SYSTEM.SAMPLE.INQ (para os programas de amostra Inquire)
- SYSTEM.SAMPLE.SET (para os programas de amostra Set)

Um fluxograma para SYSTEM.SAMPLE.ECHO é mostrado em [Figura 9 na página 1472](#) Utilizando o exemplo, o comando para emitir o pedido do programa RPG para este servidor é:

```
CALL PGM(QMQMSAMP/AMQ3REQ4) PARM('SYSTEM.SAMPLE.ECHO
+ 30 blank characters','Queue_Manager_Name')
```

porque o nome da fila e o nome do gerenciador de filas devem ter 48 caracteres de comprimento.

**Nota:** Esta fila de amostra tem um tipo de acionador de FIRST, portanto, se já houver mensagens na fila antes de executar a amostra de Solicitação, os aplicativos do servidor não serão acionados pelas mensagens que você enviar.

Se você deseja tentar exemplos adicionais, será possível tentar as variações a seguir:

- Use AMQ3TRG4 em vez de AMQ3SRV4 para enviar a tarefa em vez disso, mas os possíveis atrasos de envio de tarefa poderiam tornar menos fácil seguir o que está acontecendo
- Use o SYSTEM.SAMPLE do SYSTEM.SAMPLE.INQ e SYSTEM.SAMPLE.SET Usando o arquivo de dados de exemplo, os comandos para emitir os pedidos do programa RPG para esses servidores são:

```
CALL PGM(QMQMSAMP/AMQ3INQ4) PARM('SYSTEM.SAMPLE.INQ
+ 31 blank characters')
CALL PGM(QMQMSAMP/AMQ3SET4) PARM('SYSTEM.SAMPLE.SET
+ 31 blank characters')
```

porque o nome da fila deve ter 48 caracteres de comprimento.

Essas filas de amostra também têm um tipo de acionador de FIRST.

## ***O programa de amostra Echo em IBM i***

Os programas de amostra Echo retornam a mensagem para uma fila de resposta. O programa é denominado AMQ3ECH4

Para que o processo de acionamento funcione, você deve assegurar que o programa de amostra Echo que deseja usar seja acionado por mensagens que chegam na fila SYSTEM.SAMPLE.ECHO. Para isso, especifique o nome do programa de amostra Echo que deseja usar no campo *AppLId* da definição de processo SYSTEM.SAMPLE.ECHOPROCESS. (Para isso, é possível usar o comando CHGMQMPCR, descrito em [Administrando IBM i](#).) A fila de amostra tem um tipo de acionador FIRST, portanto, se já houver mensagens na fila antes de executar a amostra Request, a amostra Echo não será acionada pelas mensagens enviadas.

Quando você tiver configurado a definição corretamente, primeiro inicie AMQ3SRV4 em uma tarefa e, em seguida, inicie AMQ3REQ4 em outra. Você poderia usar AMQ3TRG4 em vez de AMQ3SRV4, mas possíveis atrasos de envio de tarefa poderiam tornar menos fácil seguir o que está acontecendo..

Use os programas de amostra Request para enviar mensagens para a fila SYSTEM.SAMPLE.ECHO. Os programas de amostra Echo enviam uma mensagem de resposta contendo os dados na mensagem de solicitação para a fila de resposta especificada na mensagem de solicitação.

## **Desenho do programa de amostragem Echo**

Quando o programa for acionado, ele se conectará explicitamente ao gerenciador de filas padrão usando a chamada MQCONN Embora isso não seja necessário no IBM i, isso significa que você pode utilizar o mesmo programa em outras plataformas sem alterar o código fonte.

Em seguida, o programa abre a fila nomeada na estrutura da mensagem do acionador que foi transmitida quando foi iniciada (Por questão de clareza, chamaremos isso de *fila de solicitações*.) O programa usa a chamada MQOPEN para abrir essa fila para entrada compartilhada.

O programa usa a chamada MQGET para remover as mensagens dessa fila. Esta chamada utiliza as opções GMATM e GMWT, com um intervalo de espera de 5 segundos.. O programa testa o descritor de cada mensagem para ver se é uma mensagem de solicitação; se não for, o programa descartará a mensagem e exibirá uma mensagem de aviso.

Para cada mensagem de solicitação removida da fila de solicitações, o programa usa a chamada MQPUT para colocar uma mensagem de resposta na fila de resposta.. Esta mensagem contém o conteúdo de uma mensagem de pedido.

Quando não houver nenhuma mensagem restante na fila de solicitações, o programa fecha essa fila e desconecta do gerenciador de filas.

Este programa também pode responder a mensagens enviadas para a fila de plataformas diferentes de IBM i, embora nenhuma amostra seja fornecida para esta situação. Para fazer o programa ECHO funcionar, você:

- Grave um programa, especificando corretamente os campos *Format*, *Encoding* e *CCSID* para enviar mensagens de solicitação de texto.

O programa ECHO solicita que o gerenciador de filas execute a conversão de dados da mensagem, se isto for necessário.

- Especifique CONVERT (\*YES) no canal de envio do IBM MQ for IBM i, se o programa gravado não fornecer conversão semelhante para a resposta.

### **O programa de amostra Inquire em IBM i**

O programa de amostra Inquire, AMQ3INQ4, consulta sobre alguns dos atributos de uma fila usando a chamada MQINQ

O programa deve ser executado como um programa acionado, portanto, sua única entrada é uma estrutura MQTMC (mensagem do acionador). Essa estrutura contém o nome de uma fila de destino com atributos que devem ser consultados.

Para que o processo de acionamento funcione, deve-se assegurar que o programa de amostra Inquire seja acionado por mensagens que chegam na fila SYSTEM.SAMPLE.INQ. Para fazer o ao, especifique o nome do programa de amostra Inquire no campo *AppId* do SYSTEM.SAMPLE.INQPROCESS (Para isso, é possível usar o comando CHGMQMPCRC, descrito em [Change MQ Process \(CHGMQMPCRC\)](#)). A fila de amostra tem um tipo de acionador FIRST, portanto, se já houver mensagens na fila antes de executar a amostra Request, a amostra Inquire não será acionada pelas mensagens enviadas.

Quando você tiver configurado a definição corretamente, primeiro inicie AMQ3SRV4 em uma tarefa e, em seguida, inicie AMQ3REQ4 em outra. Você poderia utilizar AMQ3TRG4 em vez de AMQ3SRV4, mas possíveis atrasos no envio de tarefa podem tornar menos fácil seguir o que está acontecendo

Use o programa de amostra Request para enviar mensagens de solicitação, cada uma contendo apenas um nome de fila, para a fila SYSTEM.SAMPLE.INQ. Para cada mensagem de solicitação, o programa de amostra Inquire envia uma mensagem de resposta que contém informações sobre a fila especificada na mensagem de solicitação. As respostas são enviadas à fila de resposta especificada na mensagem de solicitação.

### **Design do programa de amostra de consulta**

Quando o programa for acionado, ele se conectará explicitamente ao gerenciador de filas padrão usando a chamada MQCONN. Embora não seja necessário no IBM i, esse recurso de design significa que você poderia usar o mesmo programa em outras plataformas sem alterar o código fonte.

Em seguida, o programa abre a fila nomeada na estrutura da mensagem do acionador que foi transmitida quando foi iniciada (Por questão de clareza, chamaremos isso de *fila de solicitações*.) O programa usa a chamada MQOPEN para abrir essa fila para entrada compartilhada.

O programa usa a chamada MQGET para remover as mensagens dessa fila. Esta chamada utiliza as opções GMATM e GMWT, com um intervalo de espera de 5 segundos.. O programa testa o descritor de cada mensagem para ver se é uma mensagem de solicitação; se não for, o programa descarta a mensagem e exibe uma mensagem de aviso.

Para cada mensagem de solicitação removida da fila de solicitações, o programa lê o nome da fila (que chamaremos de *fila de destino*) contido nos dados e abre essa fila usando a chamada MQOPEN com a opção OOINQ. O programa então usa a chamada MQINQ para consultar sobre os valores dos atributos **InhibitGet**, **CurrentQDepth** e **OpenInputCount** da fila de destino.

Se a chamada MQINQ for bem-sucedida, o programa usa a chamada MQPUT para colocar uma mensagem de resposta na fila de resposta. Essa mensagem contém os valores dos três atributos.

Se a chamada MQOPEN ou MQINQ for malsucedida, o programa usará a chamada MQPUT para colocar uma mensagem *report* na fila de resposta. No campo *MDFB* do descritor de mensagens dessa mensagem de relatório está o código de razão retornado pela chamada MQOPEN ou MQINQ, dependendo de qual falhou.

Após a chamada MQINQ, o programa fecha a fila de destino usando a chamada MQCLOSE.

Quando não houver nenhuma mensagem restante na fila de solicitações, o programa fecha essa fila e desconecta do gerenciador de filas.

### ***O programa de amostra Set em IBM i***

O programa de amostra Set, AMQ3SET4, inibe operações put em uma fila usando a chamada MQSET para mudar o atributo **InhibitPut** da fila.

O programa é destinado a ser executado como um programa acionado, portanto, sua única entrada é uma estrutura MQTMC (mensagem do acionador) que contém o nome de uma fila de destino com atributos que devem ser consultados.

Para que o processo de acionamento funcione, deve-se assegurar que o programa de amostra Set seja acionado por mensagens que chegam na fila SYSTEM.SAMPLE.SET Para isso, especifique o nome do programa de amostra Set no campo *AppLId* da definição de processo SYSTEM.SAMPLE.SETPROCESS. (Para isso, é possível usar o comando CHGMQMPCRC, descrito em [Administrando IBM i](#).) A fila de amostra tem um tipo de acionador FIRST, portanto, se já houver mensagens na fila antes de executar a amostra Request, a amostra Set não será acionada pelas mensagens enviadas.

Quando você tiver configurado a definição corretamente, primeiro inicie AMQ3SRV4 em uma tarefa e, em seguida, inicie AMQ3REQ4 em outra. Você poderia usar AMQ3TRG4 em vez de AMQ3SRV4, mas possíveis atrasos de envio de tarefa poderiam tornar menos fácil seguir o que está acontecendo..

Use o programa de amostra Request para enviar mensagens de solicitação, cada uma contendo apenas um nome de fila, para a fila SYSTEM.SAMPLE.SET. Para cada mensagem de solicitação, o programa de amostra Set envia uma mensagem de resposta que contém uma confirmação de que as operações put foram inibidas na fila especificada As respostas são enviadas à fila de resposta especificada na mensagem de solicitação.

### **Design do programa de amostra Set**

Quando o programa for acionado, ele se conectará explicitamente ao gerenciador de filas padrão usando a chamada MQCONN Embora não seja necessário no IBM i, isso significa que você poderia usar o mesmo programa em outras plataformas sem alterar o código fonte.

Em seguida, o programa abre a fila nomeada na estrutura da mensagem do acionador que foi transmitida quando foi iniciada (Por questão de clareza, chamaremos isso de *fila de solicitações*.) O programa usa a chamada MQOPEN para abrir essa fila para entrada compartilhada.

O programa usa a chamada MQGET para remover as mensagens dessa fila. Esta chamada utiliza as opções GMATM e GMWT, com um intervalo de espera de 5 segundos.. O programa testa o descritor de cada mensagem para ver se é uma mensagem de solicitação; se não for, o programa descartará a mensagem e exibirá uma mensagem de aviso.

Para cada mensagem de solicitação removida da fila de solicitações, o programa lê o nome da fila (que chamaremos de *fila de destino*) contido nos dados e abre essa fila usando a chamada MQOPEN com a opção OOSSET. O programa então usa a chamada MQSET para configurar o valor do atributo **InhibitPut** da fila de destino para QAPUTI.

Se a chamada MQSET for bem-sucedida, o programa usa a chamada MQPUT para colocar uma mensagem de resposta na fila de resposta. Essa mensagem contém a sequência PUT inhibited.

Se a chamada MQOPEN ou MQSET for malsucedida, o programa usará a chamada MQPUT para colocar uma mensagem *report* na fila de resposta. No campo *MDFB* do descritor de mensagem desta mensagem de relatório é o código de razão retornado pela chamada MQOPEN ou MQSET, dependendo de qual falhou.

Após a chamada MQSET, o programa fecha a fila de destino usando a chamada MQCLOSE.

Quando não houver nenhuma mensagem restante na fila de solicitações, o programa fecha essa fila e desconecta do gerenciador de filas.

### **Os programas de amostra Acionamento no IBM i**

O IBM MQ for IBM i fornece dois programas de amostra de Acionamento gravados em ILE/RPG.

Os programas são:

#### **AMQ3TRG4**

Este é um monitor acionador para o ambiente do IBM i. Ele envia uma tarefa IBM i para o aplicativo ser iniciado, mas isso significa que há custo de processamento adicional associado a cada mensagem do acionador.

#### **AMQ3SRV4**

Este é um servidor acionador para o ambiente do IBM i. Para cada mensagem do acionador, esse servidor executa o comando inicial em sua própria tarefa para iniciar o aplicativo especificado. O servidor acionador pode chamar transações do CICS.

As versões de linguagem C dessas amostras também estão disponíveis como programas executáveis na biblioteca QMQM, chamada AMQSTRG4 e AMQSERV4

#### *O monitor acionador de amostra AMQ3TRG4 em IBM i*

AMQ3TRG4 é um monitor acionador.. Ele usa um parâmetro: o nome da fila de inicialização que deve servir. AMQSAMP4 define uma fila de inicialização de amostra, SYSTEM.SAMPLE.TRIGGER, que é possível usar para experimentar os programas de amostra.

AMQ3TRG4 envia uma tarefa IBM i para cada mensagem do acionador válida que obtém da fila de inicialização.

### **Concepção do monitor de disparo**

O monitor acionador abre a fila de inicialização e obtém mensagens da fila, especificando um intervalo de espera ilimitado.

O monitor acionador envia uma tarefa IBM i para iniciar o aplicativo especificado na mensagem do acionador e transmite uma estrutura MQTMC (uma versão de caractere da mensagem do acionador) Os dados do ambiente na mensagem do acionador são usados como parâmetros de envio de tarefa

Por último, o programa fecha a fila de iniciação.

#### *O servidor acionador de amostra AMQ3SRV4*

AMQ3SRV4 é um servidor acionador. Ele usa um parâmetro: o nome da fila de inicialização que deve servir. AMQSAMP4 define uma fila de inicialização de amostra, SYSTEM.SAMPLE.TRIGGER, que é possível usar para experimentar os programas de amostra.

Para cada mensagem do acionador, AMQ3SRV4 executa um comando start em sua própria tarefa para iniciar o aplicativo especificado.

Usando a fila do acionador de exemplo, o comando a ser emitido é:



```
CALL PGM(QMQM/AMQ3SRV4) PARM('Queue Name')
```

Em que Queue Name deve ter 48 caracteres de comprimento, que você consegue preenchendo o nome da fila com o número necessário de espaços em branco. Portanto, se estiver usando SYSTEM.SAMPLE.TRIGGER como sua fila de destino, você precisará de 28 caracteres em branco.

## Design do servidor acionador

O design do servidor acionador é semelhante ao do monitor acionador, exceto o servidor acionador:

- Permite CICS, bem como aplicativos IBM i
- Não usa os dados do ambiente da mensagem do acionador
- Chama aplicativos IBM i em sua própria tarefa (ou usa STRCICSUSR para iniciar os aplicativos CICS) em vez de enviar uma tarefa do IBM i
- Abre a fila de inicialização para entrada compartilhada, portanto, muitos servidores acionadores podem ser executados ao mesmo tempo

**Nota:** Os programas iniciados por AMQ3SRV4 não devem usar a chamada MQDISC porque isso parará o servidor acionador. Se os programas iniciados por AMQ3SRV4 usarem a chamada MQCONN, eles obterão o código de razão RC2002

*Finalizando os programas de amostra de acionamento no IBM i*

Um programa de monitor de acionador pode ser encerrado pela opção 2 sysrequest (ENDRQS) ou inibindo gets da fila do acionador.

Se a fila acionador de amostra for usada, o comando será:

```
CHGMQM QNAME('SYSTEM.SAMPLE.TRIGGER') GETENBL(*NO)
```

**Nota:** Para iniciar o acionamento novamente nesta fila, você deve inserir o comando:

```
CHGMQM QNAME('SYSTEM.SAMPLE.TRIGGER') GETENBL(*YES)
```

## Executando as amostras usando filas remotas no IBM i

É possível demonstrar o enfileiramento remoto executando as amostras nos gerenciadores de filas de mensagens conectados

O programa AMQSAMP4 fornece uma definição local de uma fila remota (SYSTEM.SAMPLE.REMOTE) que usa um gerenciador de filas remotas denominado OTHER. Para usar essa definição de amostra, altere OTHER para o nome do segundo gerenciador de filas de mensagens que você deseja usar. Deve-se também configurar um canal de mensagem entre seus dois gerenciadores de filas de mensagens; para obter informações sobre como fazer isso, consulte [Programas de saída do canal para canais de mensagens](#).

O programa de amostra Request coloca seu próprio nome do gerenciador de filas locais no campo MDRM de mensagens enviadas. As amostras Inquire e Set enviam mensagens de resposta para a fila e o gerenciador de filas de mensagens nomeado nos campos MDRQ e MDRM das mensagens de solicitação que eles processam

## Códigos de retorno para IBM i (ILE RPG)

Estas informações descrevem os códigos de retorno associados ao MQI e MQAI

Os códigos de retorno associados a:

- Os comandos Programmable Command Format (PCF) são listados em [Referência de formatos de comandos programáveis](#).
- As chamadas C++ são listadas em [Usando C++](#).

Para cada chamada, um código de conclusão e um código de razão são retornados pelo gerenciador de filas ou por uma rotina de saída, para indicar o sucesso ou falha da chamada.

Os aplicativos não devem depender de erros que estejam sendo verificados em uma ordem específica, exceto quando especificamente indicado. Se mais de um código de conclusão ou código de razão poderia surgir de uma chamada, o erro específico relatado depende da implementação

## Códigos de conclusão para IBM i (ILE RPG)

O parâmetro do código de conclusão (*CMPCOD*) permite que o responsável pela chamada veja rapidamente se a chamada foi concluída com êxito, parcialmente ou falhou.

### CCOK

(MQCC\_OK em outras plataformas)

Indica conclusão bem-sucedida.

A chamada foi concluída completamente; todos os parâmetros de saída foram definidos. O parâmetro **REASON** sempre tem o valor RCNONE neste caso.

### CCWARN

(MQCC\_WARN em outras plataformas)

Aviso (conclusão parcial).

A chamada foi parcialmente concluída. Alguns parâmetros de saída podem ter sido configurados, além dos parâmetros de saída *CMPCOD* e *REASON*. O parâmetro **REASON** fornece informações adicionais sobre a conclusão parcial.

### CCFAIL

(MQCC\_FAIL em outras plataformas)

A chamada falhou.

O processamento da chamada não foi concluído e o estado do gerenciador de filas normalmente é inalterado; exceções são especificamente observadas. Os parâmetros de saída *CMPCOD* e *REASON* foram configurados; outros parâmetros permanecem inalterados.

A razão pode ser uma falha no programa aplicativo ou pode ser resultado de alguma situação externa ao programa, por exemplo, a autoridade do usuário pode ter sido revogada. O parâmetro **REASON** fornece informações adicionais sobre o erro..

## Códigos de razão para IBM i (ILE RPG)

O parâmetro do código de razão (*REASON*) é uma qualificação para o parâmetro do código de conclusão (*CMPCOD*).

Se não houver nenhuma razão especial para relatar, RCNONE será retornado. Uma chamada bem-sucedida retorna CCOK e RCNONE.

Se o código de conclusão for CCWARN ou CCFAIL, o gerenciador de filas sempre relatará um motivo de qualificação; os detalhes são fornecidos sob cada descrição de chamada

Em que as rotinas de saída de usuário configuram os códigos de conclusão e razões, eles devem aderir a essas regras. Além disso, quaisquer valores de razão especiais definidos pelas saídas de usuário devem ser menores que zero, para assegurar-se de que não entrem em conflito com os valores definidos pelo gerenciador de filas. As saídas podem configurar razões já definidas pelo gerenciador de filas, quando isso for apropriado.

Códigos de razão também ocorrem em:

- O campo *DLREA* da estrutura MQDLH
- O campo *MDFB* da estrutura MQMD

Para obter a lista completa de códigos de razão, consulte [conclusão da API e códigos de razão](#).

Para localizar o seu código de razão IBM i nessa lista, remova o "RC" da frente, por exemplo, RC2002 torna-se 2002 Além disso, os códigos de conclusão lá são mostrados como eles são em outras plataformas:

<i>Tabela 814. Nomes de código de razão no IBM i e em outras plataformas..</i>	
<b>IBM i</b>	<b>Outras Plataformas</b>
CCOK	MQCC_OK
CCWARN	MQCC_WARN
CCFAIL	MQCC_FAIL

## Regras para validar opções MQI para IBM i (ILE RPG)

Este tópico fornece informações sobre as situações que produzem um código de razão RC2046 de uma chamada MQOPEN, MQPUT, MQPUT1, MQGET ou MQCLOSE.

### Chamada MQOPEN em IBM i

Para as opções da chamada MQOPEN:

- *Pelo menos um* dos seguintes deve ser especificado:

- OOB<sub>R</sub>W
- OOIN<sub>P</sub>Q
- OOIN<sub>P</sub>X
- OOIN<sub>P</sub>S
- OOIN<sub>Q</sub>
- OOOUT
- OOSET

- *Apenas um* dos seguintes é permitido:

- OOIN<sub>P</sub>Q
- OOIN<sub>P</sub>X
- OOIN<sub>P</sub>S

- *Apenas um* dos seguintes é permitido:

- OOB<sub>N</sub>D<sub>O</sub>
- OOB<sub>N</sub>D<sub>N</sub>
- OOB<sub>N</sub>D<sub>Q</sub>

**Nota:** As opções listadas anteriormente são mutuamente exclusivas Entretanto, como o valor de OOB<sub>N</sub>D<sub>Q</sub> é zero, especificá-lo com uma das outras duas opções de ligação não resulta no código de razão RC2046. OOB<sub>N</sub>D<sub>Q</sub> é fornecido para auxiliar a documentação do programa.

- Se OOSAVA for especificado, uma das opções OOIN<sub>P</sub>\* também deverá ser especificada.
- Se uma das opções OOSET\* ou OOPAS\* for especificada, OOOUT também deverá ser especificado.

### chamada MQPUT em IBM i

Para as opções put-message:

- A combinação de PMSYP e PMNSYP não é permitida
- *Apenas um* dos seguintes é permitido:
  - PMDEF<sub>C</sub>

- PMNOC
- PMPASA
- PMPASI
- PMSETA
- PMSETI
- PMALTU não é permitido (é válido apenas na chamada MQPUT1).

## **Chamada MQPUT1 em IBM i**

Para as opções put-message, as regras são as mesmas para a chamada MQPUT, exceto para as opções a seguir:

- PMALTU é permitido..
- PMLOGO não é permitido..

## **Chamada MQGET em IBM i**

Para as opções get-message:

- Apenas *uma* das seguintes opções é permitida:
  - GMNSYP
  - GMSYP
  - GMPSYP
- Apenas *uma* das seguintes opções é permitida:
  - GMBRWF
  - GMBRWC
  - GMBRWN
  - GMMUC
- GMSYP não é permitido com nenhuma das seguintes opções:
  - GMBRWF
  - GMBRWC
  - GMBRWN
  - GMLK
  - GMUNLK
- GMPSYP não é permitido com nenhuma das opções a seguir:
  - GMBRWF
  - GMBRWC
  - GMBRWN
  - GMCMPM
  - GMUNLK
- Se GMLK for especificado, uma das opções a seguir também deverá ser especificada:
  - GMBRWF
  - GMBRWC
  - GMBRWN
- Se GMUNLK for especificado, apenas as seguintes opções serão permitidas:
  - GMNSYP

- GMNWT

## Chamada MQCLOSE em IBM i

- Para as opções da chamada de MQCLOSE A combinação de CODEL e COPURG não é permitida
- Apenas um dos seguintes é permitido:
  - COKPSB
  - CORMSB

## Chamada MQSUB em IBM i

Para as opções da chamada MQSUB:

- Pelo menos um dos seguintes deve ser especificado:
- Pelo menos um dos seguintes deve ser especificado:
  - SOALT
  - SORES
  - SOCRT
- Apenas um dos seguintes é permitido:
  - SODUR
  - SONDUR

**Nota:** As opções listadas anteriormente são mutuamente exclusivas. No entanto, como o valor de SONDUR é zero, especificá-lo com SODUR não resulta no código de razão RC2046. SONDUR é fornecido para auxiliar a documentação do programa.

- A combinação SOGRP e SOMAN não é permitida.
- SOGRP requer que SOSCID seja especificado.
- Apenas um dos seguintes é permitido: SOAUID SOFUID
- A combinação de SONEWP e SOPUBR não é permitida
- SONEWP só é permitido em combinação com SOCRT.
- Apenas um dos seguintes é permitido:
  - SOWCHR
  - SOWTOP

## Codificações da máquina em IBM i

Use estas informações para aprender sobre a estrutura do campo *MDENC* no descritor de mensagem

Para obter mais informações sobre o descritor de mensagens, consulte [“MQMD \(Descritor de mensagens\) em IBM i”](#) na página 1148

O campo *MDENC* é um número inteiro de 32 bits que é dividido em quatro subcampos separados; esses subcampos identificam:

- A codificação usada para números inteiros binários
- A codificação usada para números inteiros decimais compactados
- A codificação usada para números de vírgula flutuante
- Bits reservados

Cada subcampo é identificado por uma máscara de bits que tem 1 bits nas posições correspondentes ao subcampo e 0 bits em outro lugar. Os bits são numerados de forma que o bit 0 é o bit mais significativo e o bit 31 o bit menos significativo. As máscaras a seguir são definidas:

**ENIMSK**

Máscara para codificação de número inteiro binário.

Este subcampo ocupa posições de bits 28 a 31 no campo *MDENC* .

**ENDMSK**

Máscara para codificação de número inteiro decimal compactado.

Esse subcampo ocupa as posições de bits 24 a 27 no campo *MDENC* .

**ENFMSK**

Máscara para a codificação de vírgula flutuante

Este subcampo ocupa posições de bits de 20 a 23 no campo *MDENC* .

**ENRMSK**

Máscara para bits reservados..

Esse subcampo ocupa as posições de bits de 0 a 19 no campo *MDENC*

**IBM i****Codificação de número inteiro binário em IBM i .**

Valores válidos para codificação de número inteiro binário.

Os valores a seguir são válidos para a codificação de número inteiro binário:

**ENIUND**

Codificação de número inteiro indefinida.

Os números inteiros binários são representados usando uma codificação indefinida.

**ENINOR**

Codificação de número inteiro normal.

Os números inteiros binários são representados da maneira convencional:

- O byte menos significativo no número tem o endereço mais alto de qualquer um dos bytes no número; o byte mais significativo tem o endereço mais baixo.
- O bit menos significativo em cada byte é próximo ao byte com o próximo endereço superior; o bit mais significativo em cada byte é próximo ao byte com o próximo endereço inferior.

**ENIREV**

Codificação de número inteiro revertida.

Os números inteiros binários são representados da mesma forma que ENINOR, mas com os bytes organizados em ordem inversa. Os bits dentro de cada byte são organizados da mesma forma que ENINOR.

**IBM i****Codificação de número inteiro decimal compactado em IBM i**

Valores válidos para codificação de número inteiro decimal compactado

Os valores a seguir são válidos para a codificação de número inteiro decimal compactada:

**ENDUND**

Codificação decimal compactada indefinida

Números inteiros decimais compactados são representados usando uma codificação indefinida.

**ENDNOR**

Codificação decimal compactada normal.

Números inteiros decimais compactados são representados da maneira convencional:

- Cada dígito decimal na forma de impressão do número é representado em decimal compactado por um único dígito hexadecimal no intervalo de X' 0 'a X' 9'. Cada dígito hexadecimal ocupa 4 bits e, portanto, cada byte no número decimal compactado representa dois dígitos decimais no formato imprimível do número

- O byte menos significativo no número decimal compactado é o byte que contém o dígito decimal menos significativo. Dentro desse byte, os 4 bits mais significativos contêm o dígito decimal menos significativo e os 4 bits menos significativos contêm o sinal. O sinal é 'C' (positivo), 'D' (negativo) ou 'F' (não assinado).
- O byte menos significativo no número tem o endereço mais alto de qualquer um dos bytes no número; o byte mais significativo tem o endereço mais baixo.
- O bit menos significativo em cada byte é próximo ao byte com o próximo endereço superior; o bit mais significativo em cada byte é próximo ao byte com o próximo endereço inferior.

#### **ENDREV**

Codificação de decimal compactado revertido

Números inteiros decimais compactados são representados da mesma maneira que ENDNOR, mas com os bytes organizados em ordem inversa. Os bits dentro de cada byte são organizados da mesma forma que ENDNOR.

### **IBM i Codificação de ponto flutuante em IBM i**

Valores válidos para codificação de vírgula flutuante

Os valores a seguir são válidos para a codificação de vírgula flutuante:

#### **ENFUND**

Codificação de vírgula flutuante indefinida

Os números de ponto flutuante são representados usando uma codificação indefinida.

#### **ENFNOR**

Codificação flutuante normal IEEE (The Institute of Electrical and Electronics Engineers).

Os números de vírgula flutuante são representados usando o formato de vírgula flutuante IEEE padrão, com os seguintes bytes organizados:

- O byte menos significativo na mantissa tem o endereço mais alto de qualquer um dos bytes no número; o byte que contém o expoente tem o endereço mais baixo
- O bit menos significativo em cada byte é próximo ao byte com o próximo endereço superior; o bit mais significativo em cada byte é próximo ao byte com o próximo endereço inferior

Detalhes da codificação flutuante IEEE podem ser encontrados no IEEE Padrão 754.

#### **ENFREV**

Codificação flutuante IEEE revertida.

Os números de ponto flutuante são representados da mesma maneira que ENFNOR, mas com os bytes organizados em ordem inversa. Os bits dentro de cada byte são organizados da mesma maneira que ENFNOR.

#### **ENF390**

Codificação flutuante de arquitetura do System/390

Os números de vírgula flutuante são representados usando o formato de vírgula flutuante padrão System/390 ; também é usado por System/370.

### **IBM i Construindo codificações no IBM i**

Para construir um valor para o campo *MDENC* no MQMD, as constantes relevantes que descrevem as codificações necessárias devem ser incluídas.

Certifique-se de combinar apenas uma das codificações do ENI\* com uma das codificações END\* e uma das codificações ENF\*.

## IBM i **Analisando codificações no IBM i**

O campo *MDENC* contém subcampos; por isso, os aplicativos que precisam examinar a codificação de número inteiro, decimal compactado ou flutuante devem usar a técnica descrita neste tópico.

### Usando aritmética

As etapas a seguir devem ser executadas usando aritmética de número inteiro:

1. Selecione um dos seguintes valores, de acordo com o tipo de codificação necessário:

- 1 para a codificação de número inteiro binário
- 16 para a codificação de número inteiro decimal compactado
- 256 para a codificação de ponto flutuante

Chame o valor A.

2. Divida o valor do campo *MDENC* por A ; Chame o resultado B.

3. Divida B por 16; chame o resultado de C

4. Multiplique C por 16 e subtraia de B ; Chame o resultado D.

5. Multiplique D por A ; e Chame o resultado E.

6. E é a codificação necessária e pode ser testada para igualdade com cada um dos valores válidos para esse tipo de codificação.

## IBM i **Resumo de codificações de arquitetura de máquina no IBM i .**

Uma tabela resumindo as codificações para arquiteturas de máquina

As codificações para arquiteturas de máquina são mostradas em [Tabela 815 na página 1484](#)

<b>Arquitetura da máquina</b>	<b>Codificação de número inteiro binário</b>	<b>Compactado-codificação de número inteiro decimal</b>	<b>Codificação de ponto flutuante</b>
IBM i	normal	normal	IEEE normal
Intel x86	reversed	reversed	IEEE revertido
PowerPC	normal	normal	IEEE normal
System/390	normal	normal	System/390

## IBM i **Opções de relatório e sinalizações de mensagem no IBM i**

Este tópico refere-se aos campos *MDREP* e *MDMFL* que fazem parte do descritor de mensagens MQMD especificado nas chamadas MQGET, MQPUT e MQPUT1 .

Para obter mais informações sobre o descritor de mensagens, consulte [“MQMD \(Descritor de mensagens\) em IBM i”](#) na página 1148 Estas informações descrevem:

- A estrutura do campo de relatório e como o gerenciador de filas o processa
- Como um aplicativo deve analisar o campo de relatório
- A estrutura do campo message-flags

### Estrutura do campo de relatório

O campo *MDREP* é um número inteiro de 32 bits que é dividido em três subcampos separados.

Estes subcampos identificam:



- Opções de relatório que serão rejeitadas se o gerenciador de fila local não as reconhecer
- Opções de relatório que são sempre aceitas, mesmo que o gerenciador de fila local não as reconheça
- Opções de relatório que são aceitas somente se determinadas outras condições forem satisfeitas

Cada subcampo é identificado por uma máscara de bits que tem 1 bits nas posições correspondentes ao subcampo e 0 bits em outro lugar. Observe que os bits em um subcampo não são necessariamente adjacentes. Os bits são numerados de forma que o bit 0 é o bit mais significativo e o bit 31 o bit menos significativo. As máscaras a seguir são definidas para identificar os subcampos:

#### **RORUM**

Máscara para opções de relatório não suportadas que são rejeitadas

Essa máscara identifica as posições de bits dentro do campo *MDREP* em que as opções de relatório que não são suportadas pelo gerenciador de filas locais causarão a falha da chamada MQPUT ou MQPUT1 com o código de conclusão CCFAIL e o código de razão RC2061.

Este subcampo ocupa posições de bits 3 e 11 a 13.

#### **ROAUM**

Máscara para opções de relatório não suportadas aceitas.

Essa máscara identifica as posições de bits dentro do campo *MDREP* em que as opções de relatório que não são suportadas pelo gerenciador de fila local serão, no entanto, aceitas nas chamadas MQPUT ou MQPUT1. O código de conclusão CCWARN com código de razão RC2104 é retornado neste caso.

Este subcampo ocupa posições de bits 0 a 2, 4 a 10 e 24 a 31.

As seguintes opções de relatório estão incluídas neste subcampo:

- ROCMTC
- RODLQ
- RODISC
- ROEXC
- ROEXCD
- ROEXCF
- ROEXP
- ROEXPD
- ROEXPF
- RONAN
- RONMI
- RONONE
- ROPAN
- ROPCI
- ROPMI

#### **ROAUXM**

Máscara para opções de relatório não suportadas que são aceitas apenas em certas circunstâncias

Essa máscara identifica as posições de bit dentro do campo *MDREP* no qual as opções de relatório que não são suportadas pelo gerenciador de filas locais, no entanto, serão aceitas nas chamadas MQPUT ou MQPUT1 fornecidas que ambas as condições a seguir são satisfeitas:

- A mensagem destina-se a um gerenciador de filas remotas
- O aplicativo não está colocando a mensagem diretamente em uma fila de transmissão local (ou seja, a fila identificada pelos campos *ODMN* e *ODON* no descritor de objeto especificado na chamada MQOPEN ou MQPUT1 não é uma fila de transmissão local).

O código de conclusão CCWARN com o código de razão RC2104 será retornado se essas condições forem satisfeitas e CCFAIL com o código de razão RC2061 se não.

Este subcampo ocupa posições de bits 14 a 23.

As seguintes opções de relatório estão incluídas neste subcampo:

- ROCOA
- ROCOAD
- ROCOAF
- ROCOD
- ROCODD
- ROCODF

Se houver quaisquer opções especificadas no campo *MDREP* que o gerenciador de fila não reconhece, o gerenciador de filas verificará cada subcampo por vez usando a operação AND bit a bit para combinar o campo *MDREP* com a máscara para esse subcampo. Se o resultado dessa operação não for zero, o código de conclusão e os códigos de razão descritos anteriormente serão retornados.

Se CCWARN for retornado, não será definido qual código de razão será retornado se existirem outras condições de aviso..

A capacidade de especificar e ter opções de relatório aceitas que não são reconhecidas por um gerenciador de filas locais é útil quando é necessário enviar uma mensagem com uma opção de relatório que será reconhecida e processada por um gerenciador de filas *remotas*

## **Analisando o campo de relatório em IBM i ..**

O campo MDREP contém subcampos. Por isso, alguns aplicativos precisam verificar se o emissor da mensagem solicitou um relatório específico. Esses aplicativos devem usar a técnica descrita neste tópico

### **Usando aritmética**

As etapas a seguir devem ser executadas usando aritmética de número inteiro:

1. Selecione um dos seguintes valores, de acordo com o tipo de relatório a ser verificado:

- Relatório ROCOA para COA
- Relatório ROCOD para COD
- ROEXC para relatório de exceções
- ROEXP para relatório de expiração

Chame o valor A.

2. Divida o campo *MDREP* por A ; Chame o resultado B.

3. Dividir B por 8 ; Chame o resultado C.

4. Multiplicar C por 8 e subtrair de B ; Chame o resultado D.

5. Multiplique D por A ; e Chame o resultado E.

6. Teste E para igualdade com cada um dos valores possíveis para esse tipo de relatório.

Por exemplo, se A for ROEXC, teste E para igualdade com cada um dos itens a seguir para determinar o que foi especificado pelo emissor da mensagem:

- RONONE
- ROEXC
- ROEXCD
- ROEXCF

Os testes podem ser executados em qualquer ordem que seja mais conveniente para a lógica do aplicativo

O pseudocódigo a seguir ilustra essa técnica para mensagens de relatório de exceção:

```
A = ROEXC
B = Report/A
C = B/8
D = B - C*8
E = D*A
```

Um método semelhante pode ser usado para testar as opções ROPMI ou ROPCI; selecione como o valor A qualquer um desses dois constantes é apropriado e, em seguida, continue conforme descrito anteriormente, mas substituindo o valor 8 nas etapas anteriores pelo valor 2.

## Estrutura do campo message-flags em IBM i

O campo *MDMFL* é um número inteiro de 32 bits que é dividido em três subcampos separados.

Estes subcampos identificam:

- Sinalizadores de mensagem que serão rejeitadas se o gerenciador de filas locais não os reconhecer
- Sinalizadores de mensagens que são sempre aceitos, mesmo se o gerenciador da fila local não os reconhecer
- Sinalizadores de mensagem que são aceitos apenas se determinadas outras condições forem satisfeitas..

**Nota:** Todos os subcampos em *MDMFL* são reservados para uso pelo gerenciador de fila..

Cada subcampo é identificado por uma máscara de bits que tem 1 bits nas posições correspondentes ao subcampo e 0 bits em outro lugar. Os bits são numerados de forma que o bit 0 é o bit mais significativo e o bit 31 o bit menos significativo. As máscaras a seguir são definidas para identificar os subcampos:

### **MFRUM**

Máscara para sinalizadores de mensagens não suportados que são rejeitadas

Essa máscara identifica as posições de bits dentro do campo *MDMFL* em que sinalizadores de mensagens que não são suportados pelo gerenciador de filas locais farão com que a chamada MQPUT ou MQPUT1 falhe com o código de conclusão CCFAIL e o código de razão RC2249.

Este subcampo ocupa posições de bits 20 a 31.

Os sinalizadores de mensagem a seguir estão incluídos neste subcampo:

- MFLMIG
- MFLSEG
- MFMIG
- MFSEG
- MFSEGA
- MFSEGI

### **MFAUM**

Máscara para sinalizadores de mensagem não suportados que são aceitas

Essa máscara identifica as posições de bits dentro do campo *MDMFL* em que sinalizadores de mensagens que não são suportados pelo gerenciador de filas locais serão, no entanto, aceitos nas chamadas MQPUT ou MQPUT1 . O código de conclusão é CCOK..

Este subcampo ocupa posições de bits de 0 a 11.

### **MFAUXM**

Máscara para sinalizadores de mensagem não suportados que são aceitos apenas em determinadas circunstâncias

Essa máscara identifica as posições de bits dentro do campo *MDMFL* em que os sinalizadores de mensagens que não são suportados pelo gerenciador de filas locais serão, no entanto, aceitos nas chamadas MQPUT ou MQPUT1 *fornecidas* que ambas as condições a seguir são satisfeitas:

- A mensagem destina-se a um gerenciador de filas remotas
- O aplicativo não está colocando a mensagem diretamente em uma fila de transmissão local (ou seja, a fila identificada pelos campos *ODMN* e *ODON* no descritor de objeto especificado na chamada MQOPEN ou MQPUT1 não é uma fila de transmissão local.

O código de conclusão CCOK será retornado se essas condições forem satisfeitas e CCFAIL com o código de razão RC2249 se não.

Este subcampo ocupa posições de bits 12 a 19.

Se houver sinalizadores especificados no campo *MDMFL* que o gerenciador de filas não reconhece, o gerenciador de filas verifica cada subcampo por vez usando a operação AND bit a bit para combinar o campo *MDMFL* com a máscara para esse subcampo. Se o resultado dessa operação não for zero, o código de conclusão e os códigos de razão descritos anteriormente serão retornados.

## IBM i Conversão de dados no IBM i

Este tópico descreve a interface para a saída de conversão de dados e o processamento executado pelo gerenciador de filas quando a conversão de dados é necessária..

A saída de conversão de dados é invocada como parte do processamento da chamada MQGET. Ele é usado para converter os dados da mensagem do aplicativo para a representação requerida pelo aplicativo de recebimento. A conversão dos dados da mensagem do aplicativo é opcional, e requer que a opção GMCONV seja especificada na chamada MQGET.

Os seguintes aspectos da conversão de dados são descritos:

- O processamento executado pelo gerenciador de filas em resposta para a opção GMCONV; consulte [“Processamento de conversão em IBM i”](#) na página 1488
- Convenções de processamento usadas pelo gerenciador de fila ao processar um formato integrado; essas convenções são recomendadas para saídas gravadas pelo usuário também. Consulte [“Convenções de processamento em IBM i”](#) na página 1490.
- Considerações especiais para a conversão de mensagens de relatório; consulte [“Conversão de mensagens de relatório em IBM i”](#) na página 1494
- Os parâmetros transmitidos para a saída de conversão de dados; consulte [“MQCONVX \(saída de conversão de Dados\) em IBM i”](#) na página 1505
- Uma chamada que pode ser usada a partir da saída para converter dados de caractere entre diferentes representações; consulte [“MQXCNV \(Converter caracteres\) no IBM i”](#) na página 1500
- O parâmetro de estrutura de dados que é específico para a saída; consulte [“MQDXP \(parâmetro de saída de conversão de dados\) no IBM i”](#) na página 1495

## IBM i Processamento de conversão em IBM i

Essas informações descrevem o processamento executado pelo gerenciador de filas em resposta à opção GMCONV..

O gerenciador de filas executa as ações a seguir se a opção GMCONV for especificada na chamada MQGET e houver uma mensagem a ser retornada para o aplicativo:

1. Se um ou mais dos seguintes itens for verdadeiro, nenhuma conversão será necessária:
  - Os dados da mensagem já estão no conjunto de caracteres e na codificação necessários pelo aplicativo que emite a chamada MQGET. O aplicativo deve configurar os campos *MDCSI* e *MDENC* no parâmetro **MSGDSC** da chamada MQGET para os valores necessários antes de emitir a chamada.
  - O comprimento dos dados da mensagem é zero.

- O comprimento do parâmetro **BUFFER** da chamada MQGET é zero.

Nesses casos, a mensagem é retornada sem conversão para o aplicativo emitindo a chamada MQGET; os valores *MDCSI* e *MDENC* no parâmetro **MSGDSC** são configurados para os valores nas informações de controle na mensagem e a chamada é concluída com uma das seguintes combinações de código de conclusão e código de razão:

**Código de conclusão**

**Código de razão**

**CCOK**

RCNONE

**CCWARN**

RC2079

**CCWARN**

RC2080

As etapas a seguir serão executadas somente se o conjunto de caracteres ou a codificação de dados da mensagem for diferente do valor correspondente no parâmetro **MSGDSC** e houver dados a serem convertidos:

1. Se o campo *MDFMT* nas informações de controle na mensagem tiver o valor FMNONE, a mensagem será retornada não convertida, com código de conclusão CCWARN e código de razão RC2110.

Em todos os outros casos, o processamento de conversão continua

2. A mensagem é removida da fila e colocada em um buffer temporário que tem o mesmo tamanho que o parâmetro **BUFFER**. Para operações de navegação, a mensagem é copiada no buffer temporário, em vez de ser removida da fila.
3. Se a mensagem tiver que ser truncada para caber no buffer, o seguinte será feito:
  - Se a opção GMATM não foi especificada, a mensagem será retornada sem conversão, com o código de conclusão CCWARN e o código de razão RC2080.
  - Se a opção GMATM *foi* especificada, o código de conclusão é configurado como CCWARN, o código de razão é configurado como RC2079, e o processamento de conversão continua
4. Se a mensagem puder ser acomodada no buffer sem truncamento ou a opção GMATM tiver sido especificada, o seguinte será feito:
  - Se o formato for integrado, o buffer será transmitido para o serviço de conversão de dados do gerenciador de filas.
  - Se o formato não for um formato integrado, o buffer será transmitido para uma saída gravada pelo usuário que tenha o mesmo nome que o formato. Se a saída não puder ser localizada, a mensagem será retornada sem conversão, com código de conclusão CCWARN e código de razão RC2110.

Se nenhum erro ocorrer, a saída do serviço de conversão de dados ou da saída gravada pelo usuário será a mensagem convertida, além do código de conclusão e do código de razão a ser retornado ao aplicativo que emite a chamada MQGET.

5. Se a conversão for bem-sucedida, o gerenciador de fila retorna a mensagem convertida para o aplicativo. Nesse caso, o código de conclusão e o código de razão retornados pela chamada MQGET geralmente serão uma das seguintes combinações:

**Código de conclusão**

**Código de razão**

**CCOK**

RCNONE

**CCWARN**

RC2079

No entanto, se a conversão for executada por uma saída gravada pelo usuário, outros códigos de razão poderão ser retornadas, mesmo quando a conversão for bem-sucedida

Se a conversão falhar (por qualquer motivo), o gerenciador de filas retornará a mensagem não convertida para o aplicativo, com os campos *MDCSI* e *MDENC* no parâmetro **MSGDSC** configurados para os valores nas informações de controle na mensagem e com o código de conclusão CCWARN

## IBM i Convenções de processamento em IBM i

Ao converter um formato integrado, o gerenciador de filas segue as convenções de processamento descritas neste tópico..

Considere aplicar essas convenções a saídas gravadas pelo usuário, embora isso não seja aplicado pelo gerenciador de fila Os formatos integrados convertidos pelo gerenciador de filas são os seguintes:

- FMADMN
- FMMDE
- FMCICS
- FMPCF
- FMCMD1
- FMRMH
- FMCMD2
- FMRFH
- FMDLH
- FMRFH2
- FMDH
- FMSTR
- FMEVNT
- FMTM
- FMIMS
- FMXQH
- FMIMVS

1. Se a mensagem for expandida durante a conversão e exceder o tamanho do parâmetro **BUFFER** , o seguinte será feito:
  - Se a opção GMATM não foi especificada, a mensagem será retornada sem conversão, com o código de conclusão CCWARN e o código de razão RC2120.
  - Se a opção GMATM *tiver sido* especificada, a mensagem será truncada, o código de conclusão será configurado como CCWARN, o código de razão será configurado como RC2079e o processamento de conversão continuará..
2. Se ocorrer truncamento (antes ou durante a conversão), será possível que o número de bytes válidos retornados no parâmetro **BUFFER** seja *menor que* o comprimento do buffer.

Isso pode ocorrer, por exemplo, se um número inteiro de 4 bytes ou um caractere DBCS estiver no final do buffer. O elemento incompleto de informações não é convertido e, portanto, esses bytes na mensagem retornada não contêm informações válidas Isso também pode ocorrer se uma mensagem que foi truncada antes da conversão for reduzida durante a conversão.

Se o número de bytes válidos retornados for menor que o comprimento do buffer, os bytes não utilizados no final do buffer serão configurados como nulos..

3. Se uma matriz ou cadeia transpor o fim do buffer, o maior número possível de dados será convertido; apenas o elemento de matriz ou o caractere DBCS específico que está incompleto não é convertido- os elementos ou caracteres da matriz anteriores são convertidos..
4. Se ocorrer truncamento (antes ou durante a conversão), o comprimento retornado para o parâmetro **DATLEN** será o comprimento da mensagem *não convertida* antes do truncamento.

5. Quando as cadeias são convertidas entre conjuntos de caracteres de byte único (SBCS), conjuntos de caracteres de byte duplo (DBCS) ou conjuntos de caracteres de multibyte (MBCS), as cadeias podem expandir ou contrair.

- Nos formatos de PCF FMADMN, FMEVNT e FMPCF, as sequências nas estruturas MQCFST e MQCFSL se expandem ou contratam conforme necessário para acomodar a sequência após a conversão

Para a estrutura MQCFSL da lista de sequências, as sequências na lista podem expandir ou contrair por quantias diferentes. Se isso acontecer, o gerenciador de filas preencherá as sequências mais curtas com espaços em branco para torná-las do mesmo comprimento que a sequência mais longa após a conversão

- No formato FMRMH, as sequências endereçadas pelos campos RMSEO, RMSNO, RMDEO e RMDNO se expandem ou contratam conforme necessário para acomodar as sequências após a conversão.
- No formato FMRFH, o campo RFNVS expande ou contrai conforme necessário para acomodar os pares nome-valor após a conversão.
- Em estruturas com tamanhos de campo fixos, o gerenciador de filas permite que as sequências sejam expandidas ou contradas dentro de seus campos fixos, se nenhuma informação significativa for perdida. Nesse sentido, os espaços em branco e os caracteres que seguem o primeiro caractere nulo no campo são tratados como insignificantes.
  - Se a sequência for expandida, mas apenas caracteres insignificantes precisarem ser descartados para acomodar a sequência convertida no campo, a conversão será bem-sucedida e a chamada será concluída com CCOK e código de razão RCNONE (assumindo nenhum outro erro).
  - Se a sequência for expandida, mas a sequência convertida requerer que caracteres significativos sejam descartados para se ajustarem no campo, a mensagem será retornada sem conversão e a chamada será concluída com CCWARN e código de razão RC2190.

**Nota:** O código de razão RC2190 resulta neste caso se a opção GMATM foi especificada.

- Se a sequência se contrai, o gerenciador de filas preenche a sequência com espaços em branco para o comprimento do campo.

6. Para mensagens que consistem em uma ou mais estruturas de cabeçalho IBM MQ seguidas por dados do usuário, é possível que uma ou mais das estruturas de cabeçalho sejam convertidas, enquanto o restante da mensagem não é. No entanto, com duas exceções, os campos MDCSI e MDENC em cada estrutura de cabeçalho sempre indicam corretamente o conjunto de caracteres e a codificação dos dados que seguem a estrutura do cabeçalho

As duas exceções são as estruturas MQCIH e MQIIH, em que os valores nos campos MDCSI e MDENC nessas estruturas não são significativos. Para essas estruturas, os dados após a estrutura estão no mesmo conjunto de caracteres e na codificação que a própria estrutura MQCIH ou MQIIH.

7. Se os campos MDCSI ou MDENC nas informações de controle da mensagem que está sendo recuperada ou no parâmetro **MSGDSC**, especificar valores que são indefinidos ou não suportados, o gerenciador de filas poderá ignorar o erro se o valor indefinido ou não suportado não precisar ser usado na conversão da mensagem.

Por exemplo, se o campo MDENC na mensagem especificar uma codificação flutuante não suportada, mas a mensagem contiver apenas dados de número inteiro ou contiver dados de vírgula flutuante que não requerem conversão (porque as codificações flutuantes de origem e de destino são idênticas), o erro poderá ou não ser diagnosticado.

Se o erro for diagnosticado, a mensagem será retornada sem conversão, com código de conclusão CCWARN e um dos RC2111, RC2112, RC2113, RC2114 ou RC2115, RC2116, RC2117, RC2118 códigos de razão (conforme apropriado); os campos MDCSI e MDENC no parâmetro **MSGDSC** são configurados para os valores nas informações de controle na mensagem.

Se o erro não for diagnosticado e a conversão for concluída com êxito, os valores retornados nos campos MDCSI e MDENC no parâmetro **MSGDSC** serão aqueles especificados pelo aplicativo emitindo a chamada MQGET.

8. Em todos os casos, se a mensagem for retornada ao aplicativo sem conversão, o código de conclusão será configurado como CCWARN e os campos MDCSI e MDENC no parâmetro **MSGDSC** serão configurados para os valores apropriados para os dados não convertidos. Isso é feito para FMNONE também.

O parâmetro **REASON** é configurado para um código que indica por que a conversão não pôde ser executada, a menos que a mensagem também tenha que ser truncada; os códigos de razão relacionados ao truncamento têm precedência sobre os códigos de razão relacionados à conversão (Para determinar se uma mensagem truncada foi convertida, verifique os valores retornados nos campos MDCSI e MDENC no parâmetro **MSGDSC** .)

Quando um erro é diagnosticado, um código de razão específico é retornado ou o código de razão geral RC2119.. O código de razão retornado depende dos recursos de diagnóstico do serviço de conversão de dados subjacente..

9. Se o código de conclusão CCWARN for retornado e mais de um código de razão for relevante, a ordem de precedência será a seguinte:

a. A seguinte razão tem precedência sobre todas as outras:

- RC2079

b. O próximo em precedência é o seguinte motivo:

- RC2110

c. A ordem de precedência dentro dos códigos de razão restantes não está definida

10. Na conclusão da chamada MQGET:

- O código de razão a seguir indica que a mensagem foi convertida com êxito:
  - RCNONE
- O código de razão a seguir indica que a mensagem *pode* ter sido convertida com êxito (verifique os campos MDCSI e MDENC no parâmetro **MSGDSC** para descobrir):
  - RC2079
- Todos os outros códigos de razão indicam que a mensagem não foi convertida.

O processamento a seguir é específico para os formatos integrados; ele não é aplicável a formatos definidos pelo usuário:

1. Exceto para os seguintes formatos:

- FMADMN
- FMEVNT
- FMIMVS
- FMPCF
- FMSTR

nenhum dos formatos integrados pode ser convertido de ou para conjuntos de caracteres que não possuem caracteres SBCS para os caracteres que são válidos em nomes de filas... Se for feita uma tentativa de executar essa conversão, a mensagem será retornada não convertida, com o código de conclusão CCWARN e o código de razão RC2111 ou RC2115, conforme apropriado

O conjunto de caracteres Unicode UTF-16 é um exemplo de um conjunto de caracteres que não possui caracteres SBCS para os caracteres que são válidos em nomes de fila.

2. Se os dados da mensagem para um formato integrado forem truncados, os campos na mensagem que contêm comprimentos de sequências ou contagens de elementos ou estruturas não serão ajustados para refletir o comprimento dos dados retornados ao aplicativo; os valores retornados para esses campos nos dados da mensagem são os valores aplicáveis à mensagem antes do truncamento.

Ao processar mensagens como uma mensagem FMADMN truncada, deve-se tomar cuidado para assegurar que o aplicativo não tente acessar dados além do final dos dados retornados.



3. Se o nome do formato for FMDLH, os dados da mensagem começam com uma estrutura MQDLH e podem ser seguidos por zero ou mais bytes de dados da mensagem do aplicativo. O formato, o conjunto de caracteres e a codificação dos dados da mensagem do aplicativo são definidos pelos campos DLFMT, DLCSIE DLENC na estrutura MQDLH no início de uma mensagem. Como a estrutura MQDLH e os dados da mensagem do aplicativo podem ter diferentes conjuntos de caracteres e codificações, é possível que uma, outra ou ambas as estruturas MQDLH e os dados da mensagem do aplicativo requeiram conversão.

O gerenciador de filas converte a estrutura MQDLH primeiro, conforme necessário. Se a conversão for bem-sucedida ou a estrutura MQDLH não precisar de conversão, o gerenciador de filas verificará os campos DLCSI e DLENC na estrutura MQDLH para ver se a conversão dos dados da mensagem do aplicativo é necessária. Se a conversão for necessária, o gerenciador de filas chamará a saída gravada pelo usuário com o nome fornecido pelo campo DLFMT na estrutura MQDLH ou executará a própria conversão (se DLFMT for o nome de um formato integrado).

Se a chamada MQGET retornar um código de conclusão de CCWARN e o código de razão for um daqueles que indicam que a conversão não foi bem-sucedida, um dos seguintes se aplica:

- Não foi possível converter a estrutura MQDLH. Nesse caso, os dados da mensagem do aplicativo também não terão sido convertidos
- A estrutura MQDLH foi convertida, mas os dados da mensagem do aplicativo não foram.

O aplicativo pode examinar os valores retornados nos campos MDCSI e MDENC no parâmetro **MSGDSC** e aqueles na estrutura MQDLH, para determinar qual das anteriores se aplica.

4. Se o nome do formato for FMXQH, os dados da mensagem começam com uma estrutura MQXQH e isso pode ser seguido por zero ou mais bytes de dados adicionais. Esses dados adicionais geralmente são dados da mensagem do aplicativo (que podem ter comprimento zero), mas também pode haver uma ou mais estruturas de cabeçalho IBM MQ adicionais presentes, no início dos dados adicionais.

A estrutura MQXQH deve estar no conjunto de caracteres e codificação do gerenciador de fila. O formato, o conjunto de caracteres e a codificação dos dados após a estrutura MQXQH são fornecidos pelos campos MDFMT, MDCSIE MDENC na estrutura MQMD contida no MQXQH. Para cada estrutura de cabeçalho IBM MQ subsequente presente, os campos MDFMT, MDCSIE MDENC na estrutura descrevem os dados que seguem essa estrutura; esses dados são outra estrutura de cabeçalho IBM MQ ou os dados da mensagem do aplicativo.

Se a opção GMCONV for especificada para uma mensagem FMXQH, os dados da mensagem do aplicativo e algumas das estruturas do cabeçalho MQ serão convertidas, mas os dados na estrutura MQXQH não serão. No retorno da chamada MQGET, portanto:

- Os valores dos campos MDFMT, MDCSIE MDENC no parâmetro **MSGDSC** descrevem os dados na estrutura MQXQH e não os dados da mensagem do aplicativo; portanto, os valores não serão os mesmos especificados pelo aplicativo que emitiu a chamada MQGET.

O efeito disso é que um aplicativo que obtém repetidamente mensagens de uma fila de transmissão com a opção GMCONV especificada deve reconfigurar os campos MDCSI e MDENC no parâmetro **MSGDSC** para os valores necessários para os dados da mensagens do aplicativo, antes de cada chamada MQGET

- Os valores dos campos MDFMT, MDCSIE MDENC na última estrutura do cabeçalho do MQ presente descrevem os dados da mensagem do aplicativo. Se não houver outras estruturas de cabeçalho IBM MQ presentes, os dados da mensagem do aplicativo serão descritos por esses campos na estrutura MQMD na estrutura MQXQH. Se a conversão for bem-sucedida, os valores serão os mesmos especificados no parâmetro **MSGDSC** pelo aplicativo que emitiu a chamada MQGET.

Se a mensagem for uma mensagem de lista de distribuição, a estrutura MQXQH será seguida por uma estrutura MQDH (além de suas matrizes de registros MQOR e MQPMR), que por sua vez pode ser seguida por zero ou mais estruturas de cabeçalho IBM MQ adicionais e zero ou mais bytes de dados da mensagem do aplicativo. Assim como a estrutura MQXQH, a estrutura MQDH deve estar no conjunto de caracteres e na codificação do gerenciador de filas e não é convertida na chamada MQGET, mesmo se a opção GMCONV for especificada

O processamento das estruturas MQXQH e MQDH descritas anteriormente é destinado principalmente para uso pelos agentes do canal de mensagens quando eles obtêm mensagens de filas de transmissão.

IBM i

## Conversão de mensagens de relatório em IBM i

Uma mensagem de relatório pode conter quantidades variadas de dados de mensagem do aplicativo, de acordo com as opções de relatório especificadas pelo emissor da mensagem original

Em particular, uma mensagem de relatório pode conter:

1. Nenhum dado de mensagem do aplicativo..
2. Alguns dos dados da mensagem do aplicativo da mensagem original  
Isso ocorre quando o emissor da mensagem original especifica RO\* D e a mensagem tem mais de 100 bytes.
3. Todos os dados da mensagem do aplicativo da mensagem original  
Isso ocorre quando o emissor da mensagem original especifica RO\* F ou especifica RO\* D e a mensagem tem 100 bytes ou menos.

Quando o gerenciador de filas ou o agente do canal de mensagens gera uma mensagem de relatório, ele copia o nome do formato da mensagem original no campo *MDFMT* nas informações de controle na mensagem de relatório. O nome do formato na mensagem de relatório pode, portanto, implicar um comprimento de dados diferente do comprimento presente na mensagem de relatório (casos 1 e 2 descritos anteriormente).

Se a opção GMCONV for especificada quando a mensagem de relatório for recuperada:

- Para o caso 1 descrito anteriormente, a saída de conversão de dados não será chamada (porque a mensagem de relatório não terá dados).
- Para o caso 3 descrito anteriormente, o nome do formato implica corretamente o comprimento dos dados da mensagem
- Mas para o caso 2 descrito anteriormente, a saída de conversão de dados será chamada para converter uma mensagem que seja *menor* do que o comprimento implícito pelo nome do formato.

Além disso, o código de razão transmitido para a saída geralmente será RCNONE (ou seja, o código de razão não indicará que a mensagem foi truncada).. Isso acontece porque os dados da mensagem foram truncados pelo *emissor* da mensagem de relatório, e não pelo gerenciador de filas do receptor em resposta à chamada MQGET

Devido a essas possibilidades, a saída de conversão de dados não deve usar o nome do formato para deduzir o comprimento dos dados passados para ela; em vez disso, a saída deve verificar o comprimento de dados fornecidos e estar preparada para converter menos dados do que o comprimento implícito pelo nome do formato. Se os dados puderem ser convertidos com êxito, o código de conclusão CCOK e o código de razão RCNONE devem ser retornados pela saída O comprimento dos dados da mensagem a serem convertidos é transmitido para a saída como o parâmetro **INLEN**

### Interface de programação sensível ao produto

Se uma mensagem de relatório contiver informações sobre uma atividade que ocorreu, ela será conhecida como relatório de atividades. Exemplos de atividades são:

- um MCA enviando uma mensagem de uma fila em um canal
- um MCA recebendo uma mensagem de um canal e colocando-a em uma fila
- uma mensagem de mensagens não entregues do MCA enfileirando uma mensagem não entregável
- um MCA recebendo uma mensagem de uma fila e descartando-a
- um manipulador de mensagens não entregues colocando uma mensagem novamente em uma fila
- o servidor de comandos processando uma solicitação PCF-um broker processando uma solicitação de publicação

- um aplicativo de usuário obtendo uma mensagem de uma fila-um aplicativo de usuário procurando uma mensagem em uma fila

Qualquer aplicativo, incluindo o gerenciador de filas, pode incluir alguns dos dados da mensagem no relatório de atividades após o cabeçalho do relatório.. A quantidade de dados que devem ser fornecidos se alguns forem enviados não é fixa e é decidida pelo aplicativo As informações retornadas devem ser úteis para o aplicativo que está processando o relatório de atividades Os relatórios de atividades do gerenciador de filas retornarão com eles quaisquer estruturas de cabeçalho IBM MQ padrão (iniciando 'MQH') contidas na mensagem original. Isso inclui, por exemplo, qualquer cabeçalho MQRFH2 que foi incluído na mensagem original. Também o gerenciador de filas retornará um cabeçalho MQCFH localizado, mas não os parâmetros PCF associados a ele. Isso dá aos aplicativos de monitoramento uma ideia do que era a mensagem.

IBM i

## **MQDXP (parâmetro de saída de conversão de dados) no IBM i**

Bloco do parâmetro de saída de conversões de dados

### **Visão Geral**

**Propósito:** A estrutura MQDXP é um parâmetro que o Gerenciador de Filas transmite para a saída de conversão de dados quando a saída é chamada para converter os dados da mensagem como parte do processamento da chamada MQGET Consulte a descrição da chamada MQCONVX para obter detalhes da saída de conversão de dados.

**Conjunto de caracteres e codificação:** os dados de caracteres em MQDXP estão no conjunto de caracteres do gerenciador de fila local; isso é fornecido pelo atributo do gerenciador de filas **CodedCharSetId** . Os dados numéricos em MQDXP estão na codificação da máquina nativa; isso é fornecido pelo ENNAT

**Uso:** Somente os campos *DXLEN*, *DXCC*, *DXREA* e *DXRES* em MQDXP podem ser alterados pela saída; as alterações em outros campos são ignoradas. No entanto, o campo *DXLEN* não poderá ser alterado se a mensagem que está sendo convertida for um segmento que contém apenas parte de uma mensagem lógica

Quando o controle retorna para o gerenciador de fila a partir da saída, o gerenciador de filas verifica os valores retornados em MQDXP Se os valores retornados não forem válidos, o gerenciador de filas continuará o processamento como se a saída tivesse retornado XRFAIL em *DXRES* ; no entanto, o gerenciador de filas ignora os valores dos campos *DXCC* e *DXREA* retornados pela saída nesse caso e usa em vez disso os valores que esses campos tinham na *entrada* para a saída. Os valores a seguir em MQDXP fazem com que esse processamento ocorra:

- Campo *DXRES* não XROK e não XRFAIL
- Campo *DXCC* não CCOK e não CCWARN
- *DXLEN* campo menor que zero ou *DXLEN* campo alterado quando a mensagem que está sendo convertida é um segmento que contém apenas parte de uma mensagem lógica.
- [“Campos” na página 1495](#)
- [“Declaração RPG \(copiar arquivo CMQDXPH\)” na página 1499](#)

### **Campos**

A estrutura MQDXP contém os seguintes campos; os campos são descritos em **ordem alfabética**:

#### **DXAOP (número inteiro assinado de 10 dígitos)**

Opções do aplicativo

Esta é uma cópia do campo *GMOPT* da estrutura MQGMO especificada pelo aplicativo que emite a chamada MQGET. A saída pode precisar examiná-los para verificar se a opção GMATM foi especificada

Este é um campo de entrada para a saída

### **DXCC (número inteiro assinado de 10 dígitos)**

Código de conclusão.

Quando a saída é chamada, ela contém o código de conclusão que será retornado ao aplicativo que emitiu a chamada MQGET, se a saída optar por não fazer nada. É sempre CCWARN, porque a mensagem foi truncada ou a mensagem requer conversão e isso ainda não foi feito.

Na saída da saída, esse campo contém o código de conclusão a ser retornado para o aplicativo no parâmetro **CMPCOD** da chamada MQGET; apenas CCOK e CCWARN são válidos. Consulte a descrição do campo *DXREA* para obter as sugestões sobre como a saída deve configurar esse campo na saída.

Este é um campo de entrada / saída na saída.

### **DXCSI (número inteiro assinado de 10 dígitos)**

Conjunto de caracteres requerido pelo aplicativo.

Este é o identificador do conjunto de caracteres codificado do conjunto de caracteres requerido pelo aplicativo que emite a chamada MQGET. Consulte o campo *MDCSI* na estrutura MQMD para obter mais detalhes... Se o aplicativo especificar o valor especial CSQM na chamada MQGET, o gerenciador de filas mudará isso para o identificador real do conjunto de caracteres usado pelo gerenciador de filas, antes de chamar a saída.

Se a conversão for bem-sucedida, a saída deverá copiar isso para o campo *MDCSI* no descritor de mensagens.

Este é um campo de entrada para a saída

### **DXENC (número inteiro assinado de 10 dígitos)**

Codificação numérica requerida pelo aplicativo.

Essa é a codificação numérica necessária pelo aplicativo que emite a chamada MQGET; consulte o campo *MDENC* na estrutura MQMD para obter mais detalhes.

Se a conversão for bem-sucedida, a saída deverá copiar isso para o campo *MDENC* no descritor de mensagens.

Este é um campo de entrada para a saída

### **DXHCN (número inteiro assinado de 10 dígitos)**

Manipulação de conexões.

Este é um identificador de conexão que pode ser usado na chamada MQXCNVC. Esse identificador não é necessariamente igual ao identificador especificado pelo aplicativo que emitiu a chamada MQGET.

### **DXLEN (número inteiro assinado de 10 dígitos)**

Comprimento em bytes de dados da mensagem

Quando a saída for chamada, esse campo conterá o comprimento original dos dados da mensagem do aplicativo. Se a mensagem foi truncada para se ajustar ao buffer fornecido pelo aplicativo, o tamanho da mensagem fornecida para a saída será *menor* do que o valor de *DXLEN*. O tamanho da mensagem fornecida para a saída é sempre fornecido pelo parâmetro **INLEN** da saída, independentemente de qualquer truncamento que possa ter ocorrido.

O truncamento é indicado pelo campo *DXREA* que tem o valor RC2079 na entrada para a saída..

A maioria das conversões não precisará mudar esse comprimento, mas uma saída poderá fazer isso, se necessário; o valor configurado pela saída será retornado ao aplicativo no parâmetro **DATLEN** da chamada MQGET. No entanto, este comprimento não pode ser alterado se a mensagem que está sendo convertida for um segmento que contém apenas parte de uma mensagem lógica. Isso ocorre porque alterar o comprimento faria com que os deslocamentos de segmentos posteriores na mensagem lógica fossem incorretos.

Observe que, se a saída desejar mudar o comprimento dos dados, esteja ciente de que o gerenciador de filas já decidiu se os dados da mensagem se ajustarão ao buffer do aplicativo, com base no

comprimento dos dados *não convertidos* Esta decisão determina se a mensagem é removida da fila (ou o cursor de navegação movido, para um pedido de navegação) e não é afetada por qualquer alteração no comprimento de dados causada pela conversão.. Por essa razão, recomenda-se que as saídas de conversão não causem uma mudança no comprimento dos dados da mensagem do aplicativo

Se a conversão de caracteres implicar uma mudança de comprimento, uma sequência poderá ser convertida em outra sequência com o mesmo comprimento em bytes, truncando espaços em branco à direita ou preenchendo com espaços em branco, conforme necessário,.

A saída não será invocada se a mensagem não contiver dados da mensagem do aplicativo; portanto, *DXLEN* é sempre maior que zero

Este é um campo de entrada / saída na saída.

### **DXREA (número inteiro assinado de 10 dígitos)**

Código de razão qualificando *DXCC*.

Quando a saída é chamada, ele contém o código de razão que será retornado para o aplicativo que emitiu a chamada *MQGET*, se a saída escolher não fazer nada. Entre os valores possíveis estão *RC2079*, indicando que a mensagem foi truncada para se ajustar ao buffer fornecido pelo aplicativo, e *RC2119*, indicando que a mensagem requer conversão, mas que isso ainda não foi feito

Na saída da saída, esse campo contém a razão para ser retornado ao aplicativo no parâmetro **REASON** da chamada *MQGET*; o seguinte é recomendado:

- Se *DXREA* tinha o valor *RC2079* na entrada para a saída, os campos *DXREA* e *DXCC* não devem ser alterados, independentemente de a conversão ser bem-sucedida ou falhar...

(Se o campo *DXCC* não for *CCOK*, o aplicativo que recupera a mensagem pode identificar uma falha de conversão comparando os valores *MDENC* e *MDCSI* retornados no descritor de mensagens com os valores solicitados; em contraste, o aplicativo não pode distinguir uma mensagem truncada de uma mensagem que apenas se ajustou ao buffer. Por esse motivo, *RC2079* deve ser retornado em preferência a qualquer uma das razões que indicam falha de conversão.

- Se *DXREA* tivesse qualquer outro valor na entrada para a saída:
  - Se a conversão for bem-sucedida, *DXCC* deverá ser configurado como *CCOK* e *DXREA* configurado como *RCNONE*.
  - Se a conversão falhar ou a mensagem expandir e precisar ser truncada para caber no buffer, *DXCC* deverá ser configurado como *CCWARN* (ou deixado inalterado) e *DXREA* configurado como um dos valores *i* na lista a seguir, para indicar a natureza da falha.

Observe que, se a mensagem após a conversão for muito grande para o buffer, ela deverá ser truncada apenas se o aplicativo que emitiu a chamada *MQGET* especificar a opção *GMATM*:

- Se ele especificou essa opção, a razão *RC2079* deve ser retornada
- Se ela não especificou essa opção, a mensagem deverá ser retornada sem conversão, com o código de razão *RC2120*

Os códigos de razão na lista a seguir são recomendados para uso pela saída para indicar a razão pela qual a conversão falhou, mas a saída pode retornar outros valores do conjunto de códigos *RC\**, se considerado apropriado. Além disso, o intervalo de valores *RC0900* a *RC0999* é alocado para uso pela saída para indicar condições que a saída deseja comunicar ao aplicativo que está emitindo a chamada *MQGET*.

**Nota:** Se a mensagem não puder ser convertida com êxito, a saída deverá retornar *XRFFAIL* no campo *DXRES*, para fazer com que o gerenciador de filas retorne a mensagem não convertida. Isso é verdadeiro, independentemente do código de razão retornado no campo *DXREA*

#### **RC0900**

(900, X'384 ') Valor mais baixo para o código de razão definido pelo aplicativo.

#### **RC0999**

(999, X'3E7') Valor mais alto para o código de razão definido pelo aplicativo.

**RC2120**

(2120, X'848 ') Dados convertidos muito grandes para buffer.

**RC2119**

(2119, X'847') Dados da mensagem não convertidos.

**RC2111**

(2111, X'83F') Identificador do conjunto de caracteres codificados da origem inválido.

**RC2113**

(2113, X'841') Codificação decimal compactada na mensagem não reconhecida.

**RC2114**

(2114, X'842') Codificação de ponto flutuante na mensagem não reconhecida.

**RC2112**

(2112, X'840') Codificação de número inteiro da origem não reconhecida.

**RC2115**

(2115, X'843') Identificador do conjunto de caracteres codificados do destino inválido.

**RC2117**

(2117, X'845') Codificação decimal compactada especificada pelo receptor não reconhecida.

**RC2118**

(2118, X'846') Codificação de ponto flutuante especificada pelo receptor não reconhecida.

**RC2116**

(2116, X'844') Codificação de número inteiro do destino não reconhecida.

**RC2079**

(2079, X'81F') Mensagem truncada retornada (processamento concluído).

Este é um campo de entrada / saída na saída.

**DXRES (número inteiro assinado de 10 dígitos)**

Resposta da saída.

Isto é configurado pela saída para indicar o sucesso ou não da conversão. Ele deve ser um dos seguintes:

**XR0K**

A conversão foi bem-sucedida.

Se a saída especificar esse valor, o gerenciador de filas retorna o seguinte para o aplicativo que emitiu a chamada MQGET:

- O valor do campo *DXCC* na saída da saída
- O valor do campo *DXREA* na saída da saída
- O valor do campo *DXLEN* na saída da saída
- O conteúdo do buffer de saída da saída *OUTBUF* O número de bytes retornados é o menor do parâmetro **OUTLEN** da saída e o valor do campo *DXLEN* na saída da saída

Se os campos *MDENC* e *MDCSI* no parâmetro do descritor de mensagem da saída estiverem *ambos* inalterados, o gerenciador de filas retornará:

- O valor dos campos *MDENC* e *MDCSI* na estrutura MQDXP na *entrada* para a saída

Se um ou ambos os campos *MDENC* e *MDCSI* no parâmetro do descritor de mensagem da saída tiverem sido alterados, o gerenciador de filas retornará:

- O valor dos campos *MDENC* e *MDCSI* no parâmetro do descritor de mensagem de saída na saída da saída

•

**XRFAIL**

A conversão foi malsucedida.

Se a saída especificar esse valor, o gerenciador de filas retorna o seguinte para o aplicativo que emitiu a chamada MQGET:

- O valor do campo *DXCC* na saída da saída
- O valor do campo *DXREA* na saída da saída
- O valor do campo *DXLEN* na *entrada* para a saída
- O conteúdo do buffer de entrada da saída *INBUF*.. O número de bytes retornados é fornecido pelo parâmetro **INLEN**

Se a saída alterou *INBUF*, os resultados serão indefinidos.

*DXRES* é um campo de saída da saída.

### **DXSID (sequência de caracteres de 4 bytes).**

Identificador de estruturação.

O valor deve ser:.

#### **DXSIDV**

Identificador da estrutura do parâmetro de saída de conversão de dados.

Este é um campo de entrada para a saída

### **DXVER (número inteiro assinado de 10 dígitos)**

Número de versão da estrutura.

O valor deve ser:.

#### **DXVER1**

Número da versão para a estrutura do parâmetro de saída de conversão de dados

A constante a seguir especifica o número da versão atual:

#### **DXVERC**

A versão atual da estrutura do parâmetro de saída de conversão de dados

**Nota:** Quando uma nova versão desta estrutura é introduzida, o layout da peça existente não é alterado. Portanto, a saída deve verificar se o campo *DXVER* é igual ou maior que a versão mais baixa que contém os campos que a saída precisa usar.

Este é um campo de entrada para a saída

### **DXXOP (número inteiro assinado de 10 dígitos)**

Reservado.

Este é um campo reservado; seu valor é 0.

## **Declaração RPG (copiar arquivo CMQDXPH)**

```
D*..1.....2.....3.....4.....5.....6.....7..
D* MQDXP Structure
D*
D* Structure identifier
D DXSID          1          4
D* Structure version number
D DXVER          5          8I 0
D* Reserved
D DXXOP          9          12I 0
D* Application options
D DXAOP         13          16I 0
D* Numeric encoding required by application
D DXENC         17          20I 0
D* Character set required by application
D DXCSI         21          24I 0
D* Length in bytes of message data
D DXLEN         25          28I 0
D* Completion code
```

D	DXCC	29	32I 0
D*	Reason code qualifying DXCC		
D	DXREA	33	36I 0
D*	Response from exit		
D	DXRES	37	40I 0
D*	Connection handle		
D	DXHCN	41	44I 0

## IBM i MQXCNV (Converter caracteres) no IBM i

A chamada MQXCNV converte caracteres de um conjunto de caracteres para outro.

Essa chamada faz parte da IBM MQ Data Conversion Interface (DCI), que é uma das interfaces de estrutura IBM MQ. Nota: Esta chamada pode ser usada apenas a partir de uma saída de conversão de dados..

- [“Sintaxe” na página 1500](#)
- [“Parâmetros” na página 1500](#)
- [“Chamada de RPG \(ILE\)” na página 1504](#)

### Sintaxe

**MQXCNV HCONN, OPTS, SRCCSI, SRCLEN, SRCBUF, TGTCSI, TGTLEN, TGTBUF, DATLEN, CMPCOD, REASON)**

### Parâmetros

A chamada MQXCNV possui os parâmetros a seguir:

#### HCONN (número inteiro assinado de 10 dígitos)-entrada

Manipulação de conexões.

Esta manipulação representa a conexão ao gerenciador de filas. Normalmente deve ser o identificador transmitido para a saída de conversão de dados no campo DXHCN da estrutura MQDXP; esse identificador não é necessariamente o mesmo que o identificador especificado pelo aplicativo que emitiu a chamada MQGET.

Em IBM i, o valor especial a seguir pode ser especificado para HCONN:

#### HCDEFH

Manipulação de conexões padrão.

#### OPTS (número inteiro assinado de 10 dígitos)-entrada

Opções que controlam a ação de MQXCNV.

Zero ou mais opções descritas posteriormente nesta seção podem ser especificadas. Se mais de um for necessário, os valores poderão ser adicionados (não inclua a mesma constante mais de uma vez).

**Opção de conversão padrão:** a opção a seguir controla o uso da conversão de caracteres padrão:.

#### DCCDEF

Conversão padrão..

Esta opção especifica que a conversão de caracteres padrão pode ser utilizada se um ou ambos os conjuntos de caracteres especificados na chamada não forem suportados. Isso permite que o gerenciador de filas use um conjunto de caracteres padrão especificado pela instalação que se aproxima do conjunto de caracteres especificado ao converter a sequência.

**Nota:** O resultado de usar um conjunto de caracteres aproximado para converter a sequência é que alguns caracteres podem ser convertidos incorretamente. Isso pode ser evitado usando na sequência apenas caracteres que são comuns ao conjunto de caracteres especificado e ao conjunto de caracteres padrão.



Os conjuntos de caracteres padrão são definidos por uma opção de Configuração quando o gerenciador de filas é instalado ou reiniciado.

Se DCCDEF não for especificado, o gerenciador de filas usará apenas os conjuntos de caracteres especificados para converter a sequência, e a chamada falhará se um ou ambos os conjuntos de caracteres não forem suportados

**Opção de preenchimento:** A opção a seguir permite que o gerenciador de filas preencha a sequência convertida com espaços em branco ou descarte caracteres finais insignificantes, para que a sequência convertida se ajuste ao buffer de destino:

#### **DCCFIL**

Preencha o buffer de destino

Esta opção solicita que a conversão ocorra de forma que o buffer de destino seja preenchido completamente:

- Se a sequência se contrai quando é convertida, espaços em branco à direita são incluídos para preencher o buffer de destino.
- Se a sequência for expandida quando for convertida, os caracteres finais que não forem significativos serão descartados para que a sequência convertida se ajuste ao buffer de destino. Se isso puder ser feito com êxito, a chamada será concluída com CCOK e código de razão RCNONE.

Se houver poucos caracteres finais insignificantes, tanto da sequência quanto se ajustará será colocado no buffer de destino e a chamada será concluída com CCWARN e código de razão RC2120.

Caracteres insignificantes são:

- Espaços em branco finais
- Caracteres após o primeiro caractere nulo na cadeia (mas excluindo o primeiro caractere nulo em si)
- Se a sequência, TGTCSI e TGTLEN forem tais que o buffer de destino não possa ser configurado completamente com caracteres válidos, a chamada falhará com CCFAIL e código de razão RC2144. Isso pode ocorrer quando TGTCSI é um conjunto de caracteres DBCS puro (como UTF-16), mas TGTLEN especifica um comprimento que é um número ímpar de bytes.
- TGTLEN pode ser menor que ou maior que SRCLEN No retorno de MQXCNCV, DATLEN tem o mesmo valor de TGTLEN.

Se esta opção não for especificada:.

- A sequência tem permissão para contrair ou expandir dentro do buffer de destino, conforme necessário Caracteres finais insignificantes não são incluídos ou descartados.

Se a sequência convertida se ajustar no buffer de destino, a chamada será concluída com CCOK e código de razão RCNONE.

Se a sequência convertida for muito grande para o buffer de destino, a quantidade da sequência que caberá será colocada no buffer de destino e a chamada será concluída com CCWARN e código de razão RC2120. Observe que menos de TGTLEN bytes pode ser retornado neste caso.

- TGTLEN pode ser menor que ou maior que SRCLEN No retorno de MQXCNCV, DATLEN é menor ou igual a TGTLEN.

**Opções de codificação:** As opções a seguir podem ser usadas para especificar as codificações de número inteiro das sequências de origem e destino. A codificação relevante é usada apenas quando o identificador do conjunto de caracteres correspondente indica que a representação do conjunto de caracteres no armazenamento principal depende da codificação usada para números inteiros binários. Isso afeta apenas alguns conjuntos de caracteres multibyte (por exemplo, conjuntos de caracteres UTF-16).

A codificação será ignorada se o conjunto de caracteres for um conjunto de caracteres de byte único (SBCS), ou um conjunto de caracteres multibyte com representação no armazenamento principal que não depende da codificação de número inteiro.

Apenas um dos valores DCCS\* deve ser especificado, combinado com um dos valores DCCT\*:

**DCCSNA**

Codificação de origem é o padrão para o ambiente e a linguagem de programação.

**DCCSNO**

A codificação de origem é normal

**DCCSRE**

A codificação de origem é revertida

**DCCSUN**

A codificação de origem é indefinida

**DCCTNA**

A codificação de destino é o padrão para o ambiente e linguagem de programação.

**DCCTNO**

A codificação de destino é normal

**DCCTRE**

A codificação de destino é revertida

**DCCTUN**

A codificação de destino é indefinida

Os valores de codificação definidos anteriormente podem ser incluídos diretamente no campo OPTS . No entanto, se a codificação de origem ou de destino for obtida do campo MDENC no MQMD ou em outra estrutura, o processamento a seguir deverá ser feito:

1. A codificação de número inteiro deve ser extraída do campo MDENC eliminando as codificações flutuantes e decimais compactadas; consulte [“Analisando codificações no IBM i”](#) na página 1484 para obter detalhes de como fazer isso.
2. A codificação de número inteiro resultante da etapa 1 deve ser multiplicada pelo fator apropriado antes de ser incluída no campo OPTS .. Esses fatores são:

**DCCSFA**

Fator para codificação de origem

**DCCTFA**

Fator para codificação de destino

Se não especificado, as opções de codificação serão padronizadas como indefinidas (DCC\* UN). Na maioria dos casos, isso não afeta a conclusão bem-sucedida da chamada MQXCNV.. No entanto, se o conjunto de caracteres correspondente for um conjunto de caracteres multibyte com representação que é dependente da codificação (por exemplo, um conjunto de caracteres UTF-16 ), a chamada falhará com o código de razão RC2112 ou RC2116 conforme apropriado.

**Opção padrão:** Se nenhuma das opções descritas anteriormente for especificada a seguinte opção pode ser usada:

**DCCNON**

Nenhuma opção especificada.

O DCCNON é definido para auxiliar a documentação do programa. Não se pretende que essa opção seja usada com qualquer outra, mas como seu valor é zero, tal uso não pode ser detectado.

**SRCCSI (número inteiro assinado de 10 dígitos)-entrada**

Identificador do conjunto de caracteres codificados da cadeia antes da conversão.

Esse é o identificador de conjunto de caracteres codificados da sequência de entrada em SRCBUF.

**SRCLLEN (número inteiro assinado de 10 dígitos)-entrada**

Comprimento da sequência antes da conversão..

Esse é o comprimento em bytes da sequência de entrada em SRCBUF ; deve ser zero ou maior.

**SRCBUF (sequência de caracteres de 1 byte x SRCLEN)-entrada**

Sequência a ser convertida.

Este é o buffer que contém a cadeia a ser convertida de um conjunto de caracteres para outro

**TGTCSI (número inteiro assinado de 10 dígitos)-entrada**

Identificador do conjunto de caracteres codificados da cadeia após conversão.

Este é o identificador do conjunto de caractere codificado do conjunto de caracteres para o qual SRCBUF deve ser convertido.

**TGTLEN (número inteiro assinado de 10 dígitos)-entrada**

Comprimento do buffer de saída..

Esse é o comprimento em bytes do buffer de saída TGTBUF ; deve ser zero ou maior. Pode ser menor ou maior que SRCLEN.

**TGTBUF (cadeia de caracteres de 1 byte x TGTLEN)-saída.**

Sequência após a conversão

Esta é a sequência após ela ter sido convertida para o conjunto de caracteres definido por TGTCSI A sequência convertida pode ser menor ou maior que a sequência não convertida. O parâmetro **DATLEN** indica o número de bytes válidos retornados

**DATLEN (número inteiro assinado de 10 dígitos)-saída**

Comprimento da sequência de saída

Este é o comprimento da sequência retornada no buffer de saída TGTBUF A sequência convertida pode ser menor ou maior que a sequência não convertida.

**CMPCOD (número inteiro assinado de 10 dígitos)-saída**

Código de conclusão.

É um dos seguintes:

**CCOK**

Indica conclusão bem-sucedida.

**CCWARN**

Aviso (conclusão parcial).

**CCFAIL**

A chamada falhou.

**REASON (número inteiro assinado de 10 dígitos)-saída**

Código de razão qualificando CMPCOD.

Se CMPCOD for CCOK:

**RCNONE**

(0, X'000') Nenhuma razão a relatar.

Se CMPCOD for CCWARN:

**RC2120**

(2120, X'848') Dados convertidos muito grandes para o buffer.

Se CMPCOD for CCFAIL:

**RC2010**

(2010, X'7DA') Parâmetro de comprimento de dados inválido.

**RC2150**

(2150, X'866') Sequência DBCS inválida.

**RC2018**

(2018, X'7E2') Manipulação de conexões não válida.

**RC2046**

(2046, X'7FE') Opções não válidas ou não consistentes.

**RC2102**

(2102, X'836') Recursos insuficientes do sistema disponíveis.

**RC2145**

(2145, X'861') Parâmetro de buffer de origem inválido.

**RC2111**

(2111, X'83F') Identificador do conjunto de caracteres codificados da origem inválido.

**RC2112**

(2112, X'840') Codificação de número inteiro da origem não reconhecida.

**RC2143**

(2143, X'85F') Parâmetro de comprimento de origem inválido.

**RC2071**

(2071, X'817') Armazenamento insuficiente disponível.

**RC2146**

(2146, X'862') Parâmetro de buffer de destino inválido.

**RC2115**

(2115, X'843') Identificador do conjunto de caracteres codificados do destino inválido.

**RC2116**

(2116, X'844') Codificação de número inteiro do destino não reconhecida.

**RC2144**

(2144, X'860 ') O parâmetro de comprimento de destino não é válido.

**RC2195**

(2195, X'893') Ocorreu um erro inesperado.

Para obter mais informações sobre esses códigos de razão, consulte [“Códigos de retorno para IBM i \(ILE RPG\)”](#) na página 1477

**Chamada de RPG (ILE)**

```

C*..1.....2.....3.....4.....5.....6.....7..
C          CALLP      MQXCNVC(HCONN : OPTS : SRCCSI :
C                               SRCLen : SRCBUF : TGTCSE :
C                               TGTLEN : TGTBUF : DATLEN :
C                               CMPCOD : REASON)

```

A definição de protótipo para a chamada é:

```

D*..1.....2.....3.....4.....5.....6.....7..
DMQXCNVC      PR          EXTPROC('MQXCNVC')
D* Connection handle
D HCONN              10I 0 VALUE
D* Options that control the action of MQXCNVC
D OPTS              10I 0 VALUE
D* Coded character set identifier of string before conversion
D SRCCSI            10I 0 VALUE
D* Length of string before conversion
D SRCLen            10I 0 VALUE
D* String to be converted
D SRCBUF            *   VALUE
D* Coded character set identifier of string after conversion
D TGTCSE            10I 0 VALUE
D* Length of output buffer
D TGTLEN            10I 0 VALUE
D* String after conversion
D TGTBUF            *   VALUE
D* Length of output string
D DATLEN            10I 0

```

D* Completion code	
D CMPCOD	10I 0
D* Reason code qualifying CMPCOD	
D REASON	10I 0

## IBM i **MQCONVX (saída de conversão de Dados) em IBM i**

Esta definição de chamada descreve os parâmetros que são passados para a saída de conversão de dados

Nenhum ponto de entrada chamado MQCONVX é fornecido pelo gerenciador de filas (consulte a nota de uso “11” na página 1507).

Esta definição faz parte da IBM MQ Data Conversion Interface (DCI), que é uma das interfaces de estrutura do IBM MQ .

- “Sintaxe” na página 1505
- “Observações de Uso” na página 1505
- “Parâmetros” na página 1507
- “Chamada de RPG (ILE)” na página 1508

### Sintaxe

**MQCONVX (MQDXP, MQMD, INLEN, INBUF, OUTLEN, OUTBUF)**

### Observações de Uso

1. Uma saída de conversão de dados é uma saída gravada pelo usuário que recebe o controle durante o processamento de uma chamada MQGET. A função executada pela saída de conversão de dados é definida pelo provedor da saída; no entanto, a saída deve estar em conformidade com as regras descritas aqui e na estrutura de parâmetro MQDXP associada.

As linguagens de programação que podem ser usadas para uma saída de conversão de dados são determinadas pelo ambiente.

2. A saída será chamada apenas se *todas* as seguintes instruções forem verdadeiras:
  - A opção GMPCONV é especificada na chamada MQGET..
  - O campo *MDFMT* no descritor de mensagens não é FMNONE.
  - A mensagem ainda não está na representação necessária; ou seja, um ou ambos os *MDCSI* e *MDENC* da mensagem são diferentes do valor especificado pelo aplicativo no descritor de mensagens fornecido na chamada MQGET
  - O gerenciador de filas ainda não fez a conversão com êxito
  - O comprimento do buffer do aplicativo é maior que zero
  - O comprimento dos dados da mensagem é maior que zero
  - O código de razão até agora durante a operação MQGET é RCNONE ou RC2079
3. Quando uma saída está sendo gravada, deve-se considerar a codificação da saída de uma maneira que permita converter mensagens que foram truncadas. Mensagens truncadas podem surgir das seguintes maneiras:
  - O aplicativo de recebimento fornece um buffer menor que a mensagem, mas especifica a opção GMATM na chamada MQGET.

Neste caso, o campo *DXREA* no parâmetro **MQDXP** na entrada para a saída terá o valor RC2079.
  - O emissor da mensagem a truncou antes de enviá-la. Isso pode acontecer com mensagens de relatório, por exemplo (consulte “Conversão de mensagens de relatório em IBM i” na página 1494 para obter mais detalhes)

Nesse caso, o campo *DXREA* no parâmetro **MQDXP** na entrada para a saída terá o valor RCNONE (se o aplicativo de recebimento tiver fornecido um buffer que era grande o suficiente para a mensagem)

Portanto, o valor do campo *DXREA* na entrada para a saída nem sempre pode ser usado para decidir se a mensagem foi truncada.

A característica distinta de uma mensagem truncada é que o comprimento fornecido para a saída no parâmetro **INLEN** será *menor que* o comprimento implícito pelo nome do formato contido no campo *MDFMT* no descritor de mensagens. Portanto, a saída deve verificar o valor de *INLEN* antes de tentar converter qualquer um dos dados; a saída *não deve* assumir que a quantia total de dados implícita pelo nome do formato foi fornecida.

Se a saída não tiver sido gravada para converter mensagens truncadas e **INLEN** for menor que o valor esperado, a saída deverá retornar XRFAIL no campo *DXRES* do parâmetro **MQDXP**, com o campo *DXCC* configurado como CCWARN e o campo *DXREA* configurado como RC2110.

Se a saída *tiver* sido gravada para converter mensagens truncadas, a saída deverá converter o máximo de dados possível (consulte a próxima nota de uso), tomando cuidado para não tentar examinar ou converter dados além do término de *INBUF*. Se a conversão for concluída com êxito, a saída deverá deixar o campo *DXREA* no parâmetro **MQDXP** inalterado. Isso retornará RC2079 se a mensagem foi truncada pelo gerenciador de filas do receptor e RCNONE se a mensagem foi truncada pelo emissor da mensagem.

Também é possível para uma mensagem expandir *durante* conversão, para o ponto em que ele é maior que *OUTBUF*. Nesse caso, a saída deve decidir se truncará a mensagem; o campo *DXAOP* do parâmetro **MQDXP** indicará se o aplicativo de recebimento especificou a opção GMATM

4. Geralmente, é recomendado que todos os dados na mensagem fornecida para a saída em *INBUF* sejam convertidos ou que nenhum deles seja. Uma exceção a isso, no entanto, ocorre se a mensagem for truncada, antes da conversão ou durante a conversão; neste caso, pode haver um item incompleto no final do buffer (por exemplo: um byte de um caractere de byte duplo ou 3 bytes de um número inteiro de 4 bytes). Nessa situação, é recomendado que o item incompleto seja omitido e bytes não usados em *OUTBUF* configurados como nulos. No entanto, elementos ou caracteres completos em uma matriz ou sequência *devem* ser convertidos.
5. Quando uma saída for necessária pela primeira vez, o gerenciador de filas tentará carregar um objeto que tenha o mesmo nome que o formato (além de extensões). O objeto carregado deve conter a saída que processa mensagens com esse nome de formato.. É recomendado que o nome da saída e o nome do objeto que contém a saída sejam idênticos, embora nem todos os ambientes requeiram isso.
6. Uma nova cópia da saída é carregada quando um aplicativo tenta recuperar a primeira mensagem que usa esse *MDFMT* desde o aplicativo conectado ao gerenciador de filas. Uma nova cópia também pode ser carregada em outros momentos, se o gerenciador de filas tiver descartado uma cópia carregada anteriormente. Por essa razão, uma saída não deve tentar usar armazenamento estático para comunicar informações de uma chamada da saída para a próxima-a saída pode ser descarregada entre as duas chamadas.
7. Se houver uma saída fornecida pelo usuário com o mesmo nome que um dos formatos integrados suportados pelo gerenciador de fila, a saída fornecida pelo usuário não substituirá a rotina de conversão integrada. As únicas circunstâncias em que tal saída é invocada são:
  - Se a rotina de conversão integrada não puder manipular conversões para ou a partir do *MDCSI* ou *MDENC* envolvido, ou
  - Se a rotina de conversão integrada falhou ao converter os dados (por exemplo, porque há um campo ou caractere que não pode ser convertido).
8. O escopo da saída é dependente do ambiente. Os nomes de *MDFMT* devem ser escolhidos para minimizar o risco de conflitos com outros formatos. É recomendado que eles comecem com caracteres que identifiquem o aplicativo que define o nome do formato.
9. A saída de conversão de dados é executada em um ambiente como aquele do programa que emitiu a chamada MQGET; o ambiente inclui espaço de endereço e perfil do usuário (onde aplicável). O

programa pode ser um agente do canal de mensagens enviando mensagens para um gerenciador de filas de destino que não suporta conversão de mensagens. A saída não pode comprometer a integridade do gerenciador de filas, pois ele não é executado no ambiente do gerenciador de filas.

10. A única chamada MQI que pode ser usada pela saída é MQXCNVC; tentar usar outras chamadas MQI falha com o código de razão RC2219 ou outros erros imprevisíveis.
11. Nenhum ponto de entrada chamado MQCONVX é fornecido pelo gerenciador de filas. O nome da saída deve ser igual ao nome do formato (o nome contido no campo *MDFMT* em MQMD), embora isso não seja necessário em todos os ambientes.

## Parâmetros

A chamada MQCONVX possui os parâmetros a seguir:

### MQDXP (MQDXP)-entrada/saída

Bloco do parâmetro de saída de conversões de dados

Essa estrutura contém informações relacionadas à invocação da saída. A saída configura as informações nessa estrutura para indicar o resultado da conversão. Consulte “MQDXP (parâmetro de saída de conversão de dados) no IBM i” na página 1495 para obter detalhes dos campos nesta estrutura..

### MQMD (MQMD)-entrada/saída

Descritor de mensagens

Na entrada para a saída, este é o descritor de mensagem que seria retornado para o aplicativo se nenhuma conversão fosse executada. Portanto, ele contém o *MDFMT*, *MDENC* e *MDCSI* da mensagem não convertida contida em *INBUF*.

**Nota:** O parâmetro **MQMD** passado para a saída é sempre a versão mais recente do MQMD suportada pelo gerenciador de fila que chama a saída. Se a saída for destinada a ser móvel entre diferentes ambientes, a saída deverá verificar o campo *MDVER* em *MQMD* para verificar se os campos que a saída precisa acessar estão presentes na estrutura.

No IBM i, a saída é transmitida um MQMD version-2.

Na saída, a saída deve alterar os campos *MDENC* e *MDCSI* para os valores solicitados pelo aplicativo, se a conversão foi bem-sucedida; essas mudanças serão refletidas de volta ao aplicativo. Quaisquer outras mudanças que a saída faz na estrutura são ignoradas; elas não são refletidas de volta para o aplicativo.

Se a saída retornar *XROK* no campo *DXRES* da estrutura MQDXP, mas não alterar os campos *MDENC* ou *MDCSI* no descritor de mensagem, o gerenciador de filas retornará para esses campos os valores que os campos correspondentes na estrutura MQDXP tinham na entrada para a saída.

### INLEN (número inteiro assinado de 10 dígitos)-entrada

Comprimento em bytes de *INBUF*.

Esse é o comprimento do buffer de entrada *INBUF* e especifica o número de bytes a serem processados pela saída. *INLEN* é o menor comprimento dos dados da mensagem antes da conversão e o comprimento do buffer fornecido pelo aplicativo na chamada MQGET.

O valor é sempre maior do que zero.

### INBUF (1 byte bit string x INLEN)-entrada

Buffer contendo a mensagem não convertida.

Isso contém os dados da mensagem antes da conversão. Se a saída não puder converter os dados, o gerenciador de filas retornará o conteúdo desse buffer para o aplicativo após a conclusão da saída.

**Nota:** A saída não deve alterar *INBUF*; Se esse parâmetro for alterado, os resultados serão indefinidos.

## OUTLEN (número inteiro assinado de 10 dígitos)-entrada

Comprimento em bytes de *OUTBUF*.

Esse é o comprimento do buffer de saída *OUTBUF* e é o mesmo comprimento do buffer fornecido pelo aplicativo na chamada *MQGET*

O valor é sempre maior do que zero.

## OUTBUF (1 byte bit string x OUTLEN)-saída

Buffer contendo a mensagem convertida.

Na saída da saída, se a conversão foi bem-sucedida (conforme indicado pelo valor *XROK* no campo *DXRES* do parâmetro **MQDXP**), **OUTBUF** contém os dados da mensagem a serem entregues ao aplicativo, na representação solicitada. Se a conversão foi malsucedida, quaisquer alterações que a saída tenha feito para este buffer serão ignorados

## Chamada de RPG (ILE)

```
C*..1.....2.....3.....4.....5.....6.....7..
C          CALLP      exitname(MQDXP : MQMD : INLEN :
C                               INBUF : OUTLEN : OUTBUF)
```

A definição de protótipo para a chamada é:

```
D*..1.....2.....3.....4.....5.....6.....7..
Dexitname      PR          EXTPROC('exitname')
D* Data-conversion exit parameter block
D MQDXP                44A
D* Message descriptor
D MQMD                  364A
D* Length in bytes of INBUF
D INLEN                 10I 0 VALUE
D* Buffer containing the unconverted message
D INBUF                 *  VALUE
D* Length in bytes of OUTBUF
D OUTLEN                10I 0 VALUE
D* Buffer containing the converted message
D OUTBUF                *  VALUE
```

**Fim da interface de programação sensível ao produto**

## Saídas de usuário, saídas de API e referência de serviços instaláveis

Use as informações nesta seção para ajudá-lo a desenvolver suas saídas de usuário, saídas de API e aplicativos de serviços instaláveis:

- [“Estrutura MQIEP” na página 1509](#)
- [“Referência de saída de conversão de dados” na página 1512](#)
- [“MQ\\_PUBLISH\\_EXIT - saída Publish” na página 1516](#)
- [“Chamadas de Saída do Canal e Estrutura de Dados” na página 1524](#)
- [“Referência de saída de API” na página 1616](#)
- [“Informações de referência da interface de serviços instaláveis” na página 1678](#)

### Conceitos relacionados

[Saídas de usuário, saídas de API e serviços instaláveis do IBM MQ](#)

### Tarefas relacionadas

[Estendendo as instalações do gerenciador de filas](#)



## Estrutura MQIEP

A estrutura MQIEP contém um ponto de entrada para cada chamada de função que as saídas podem fazer.

### Campos

#### StrucId

Tipo: MQCHAR4 - entrada

Identificador de estruturação. O valor é o seguinte:

#### **MQIEP\_STRUC\_ID**

#### Versão

Tipo: MQLONG - entrada

Número de versão da estrutura. O valor é o seguinte:

#### **MQIEP\_VERSION\_1**

Número da versão da estrutura da versão 1

#### **MQIEP\_CURRENT\_VERSION**

A versão atual da estrutura.

#### StrucLength

Tipo: MQLONG

Tamanho da estrutura MQIEP em bytes.. O valor é o seguinte:

#### **MQIEP\_LENGTH\_1**

#### Sinalizadores

Tipo: MQLONG

Fornece informações sobre os endereços de função Um sinalizador para indicar se a biblioteca está encadeada pode ser usado com um sinalizador para indicar se a biblioteca é uma biblioteca do cliente ou do servidor

O valor a seguir é usado para especificar nenhuma informação de biblioteca:.

#### **MQIEPF\_NONE**

Um dos seguintes valores é utilizado para especificar se a biblioteca compartilhada é encadeada ou não encadeada:

#### **MQIEPF\_NON\_THREADED\_LIBRARY**

Uma biblioteca compartilhada não encadeada

#### **MQIEPF\_THREADED\_LIBRARY**

Uma biblioteca compartilhada encadeada

Um dos seguintes valores é usado para especificar se a biblioteca compartilhada é um cliente ou uma biblioteca compartilhada do servidor:

#### **MQIEPF\_CLIENT\_LIBRARY**

Uma biblioteca compartilhada do cliente

#### **MQIEPF\_LOCAL\_LIBRARY**

Uma biblioteca compartilhada do servidor

#### Reservado

Tipo: MQPTR

#### MQBACK\_Call

Tipo: PMQ\_BACK\_CALL

Endereço da chamada MQBACK.

**MQBEGIN\_Call**

Tipo: PMQ\_BEGIN\_CALL

Endereço da chamada de MQBEGIN

**MQBUFMH\_Call**

Tipo: PMQ\_BUFMH\_CALL

Endereço da chamada de MQBUFMH

**MQCB\_Call**

Tipo: PMQ\_CB\_CALL

Endereço da chamada de MQCB

**MQCLOSE\_Call**

Tipo: PMQ\_CLOSE\_CALL

Endereço da chamada de MQCLOSE

**MQCMIT\_Call**

Tipo: PMQ\_CMIT\_CALL

Endereço da chamada MQCMIT.

**MQCONN\_Call**

Tipo: PMQ\_CONN\_CALL

Endereço da chamada MQCONN.

**MQCONNX\_Call**

Tipo: PMQ\_CONNX\_CALL

Endereço da chamada MQCONNX..

**MQCRTMH\_Call**

Tipo: PMQ\_CRTMH\_CALL

Endereço da chamada de MQCRTMH

**MQCTL\_Call**

Tipo: PMQ\_CTL\_CALL

Endereço da chamada de MQCTL

**MQDISC\_Call**

Tipo: PMQ\_DISC\_CALL

Endereço da chamada de MQDISC

**MQDLTMH\_Call**

Tipo: PMQ\_DLTMH\_CALL

Endereço da chamada de MQDLTMH

**MQDLTMP\_Call**

Tipo: PMQ\_DLTMP\_CALL

Endereço da chamada MQDLTMP..

**MQGET\_Call**

Tipo: PMQ\_GET\_CALL

Endereço da chamada MQGET..

**MQINQ\_Call**

Tipo: PMQ\_INQ\_CALL

Endereço da chamada de MQINQ

**MQINQMP\_Call**

Tipo: PMQ\_INQMP\_CALL

Endereço da chamada de MQINQMP

**MQMHBUF\_Call**

Tipo: PMQ\_MHBUF\_CALL

Endereço da chamada de MQMHBUF

**MQOPEN\_Call**

Tipo: PMQ\_OPEN\_CALL

Endereço da chamada MQOPEN.

**MQPUT\_Call**

Tipo: PMQ\_PUT\_CALL

Endereço da chamada MQPUT..

**MQPUT1\_Call**

Tipo: PMQ\_PUT1\_CALL

Endereço da chamada MQPUT1 .

**MQSET\_Call**

Tipo: PMQ\_SET\_CALL

Endereço da chamada de MQSET

**MQSETMP\_Call**

Tipo: PMQ\_SETMP\_CALL

Endereço da chamada MQSETMP.

**MQSTAT\_Call**

Tipo: PMQ\_STAT\_CALL

Endereço da chamada de MQSTAT

**MQSUB\_Call**

Tipo: PMQ\_SUB\_CALL

Endereço da chamada MQSUB.

**MQSUBRQ\_Call**

Tipo: PMQ\_SUBRQ\_CALL

Endereço da chamada MQSUBRQ.

**MQXCNVC\_Call**

Tipo: PMQ\_XCNVC\_CALL

Endereço da chamada de MQXCNVC

**MQXCLWLN\_Call**

Tipo: PMQ\_XCLWLN\_CALL

Endereço da chamada MQXCLWLN.

**MQXDX\_Call**

Tipo: PMQ\_XDX\_CALL

Endereço da chamada MQXDX.

**MQXEP\_Call**

Tipo: PMQ\_XEP\_CALL

Endereço da chamada MQXEP.

**MQZEP\_Call**

Tipo: PMQ\_ZEP\_CALL

Endereço da chamada de MQZEP

## C Declaração

```
struct tagMQIEP {
    MQCHAR4      StrucId;          /* Structure identifier */
    MQLONG       Version;         /* Structure version number */
    MQLONG       StrucLength;     /* Structure length */
    MQLONG       Flags;           /* Flags */
    MQPTR        Reserved;       /* Reserved */
    PMQ_BACK_CALL MQBACK_Call;    /* Address of MQBACK */
    PMQ_BEGIN_CALL MQBEGIN_Call;  /* Address of MQBEGIN */
    PMQ_BUFMH_CALL MQBUFMH_Call;  /* Address of MQBUFMH */
    PMQ_CB_CALL  MQCB_Call;       /* Address of MQCB */
    PMQ_CLOSE_CALL MQCLOSE_Call;  /* Address of MQCLOSE */
    PMQ_CMIT_CALL MQCMIT_Call;    /* Address of MQCMIT */
    PMQ_CONN_CALL MQCONN_Call;    /* Address of MQCONN */
    PMQ_CONNX_CALL MQCONNX_Call;  /* Address of MQCONNX */
    PMQ_CRTMH_CALL MQCRTMH_Call;  /* Address of MQCRTMH */
    PMQ_CTL_CALL  MQCTL_Call;     /* Address of MQCTL */
    PMQ_DISC_CALL MQDISC_Call;    /* Address of MQDISC */
    PMQ_DLTMH_CALL MQDLTMH_Call;  /* Address of MQDLTMH */
    PMQ_DLTMP_CALL MQDLTMP_Call;  /* Address of MQDLTMP */
    PMQ_GET_CALL  MQGET_Call;     /* Address of MQGET */
    PMQ_INQ_CALL  MQINQ_Call;     /* Address of MQINQ */
    PMQ_INQMP_CALL MQINQMP_Call;  /* Address of MQINQMP */
    PMQ_MHBUF_CALL MQMHBUF_Call;  /* Address of MQMHBUF */
    PMQ_OPEN_CALL MQOPEN_Call;    /* Address of MQOPEN */
    PMQ_PUT_CALL  MQPUT_Call;     /* Address of MQPUT */
    PMQ_PUT1_CALL MQPUT1_Call;    /* Address of MQPUT1 */
    PMQ_SET_CALL  MQSET_Call;     /* Address of MQSET */
    PMQ_SETMP_CALL MQSETMP_Call;  /* Address of MQSETMP */
    PMQ_STAT_CALL MQSTAT_Call;    /* Address of MQSTAT */
    PMQ_SUB_CALL  MQSUB_Call;     /* Address of MQSUB */
    PMQ_SUBRQ_CALL MQSUBRQ_Call;  /* Address of MQSUBRQ */
    PMQ_XCLWLN_CALL MQXCLWLN_Call; /* Address of MQXCLWLN */
    PMQ_XCNVC_CALL MQXCNVC_Call;  /* Address of MQXCNVC */
    PMQ_XDX_CALL  MQXDX_Call;     /* Address of MQXDX */
    PMQ_XEP_CALL  MQXEP_Call;     /* Address of MQXEP */
    PMQ_ZEP_CALL  MQZEP_Call;     /* Address of MQZEP */
};
```



## Referência de saída de conversão de dados

Para z/OS, você deve gravar saídas de conversão de dados em linguagem assembler. Para outras plataformas, é recomendado usar a linguagem de programação C.

Para ajudá-lo a criar um programa de saída de conversão de dados, os seguintes recursos são fornecidos:

- Um arquivo de origem de estrutura básica
- Uma chamada de caracteres convertidos
- Um utilitário que cria um fragmento de código que executa conversão de dados em estruturas de tipo de dados Este utilitário usa apenas entrada C. No z/OS, ele produz código assembler.

Para obter o procedimento para gravar os programas, consulte:






-  [Gravando um programa de saída de conversão de dados para IBM MQ for IBM i](#)
-  [Gravando um programa de saída de conversão de dados para IBM MQ for z/OS](#)
- [Gravando uma saída de conversão de dados para IBM MQ for AIX or Linux sistemas](#)
- [Gravando uma saída de conversão de dados para IBM MQ for Windows](#)

## Arquivo de origem da estrutura básica

Estes podem ser usados como seu ponto de início ao gravar um programa de saída de conversão de dados.

Os arquivos fornecidos são listados em [Tabela 816 na página 1513](#)

Tabela 816. Arquivos de origem de estrutura básica

Plataforma	Arquivo
 AIX	amqsvfc0.c
 IBM i	QMQMSAMP/QCSRC (AMQSVFC4)
 Linux	amqsvfc0.c
Sistemas  Windows	amqsvfc0.c
 z/OS	CSQ4BAX8 ( “1” na página 1513 ) CSQ4BAX9 ( “2” na página 1513 ) CSQ4CAX9 ( “3” na página 1513 )
<b>Notas:</b>	
<ol style="list-style-type: none"> <li>1. Ilustra a chamada de MQXCVNC</li> <li>2. Um wrapper para os fragmentos de código gerados pelo utilitário para uso em todos os ambientes, exceto CICS..</li> <li>3. Um wrapper para os fragmentos de código gerados pelo utilitário para uso no ambiente do CICS</li> </ol>	

## Converter chamada de caracteres

Use a chamada MQXCNV (converter caracteres) de dentro de um programa de saída de conversão de dados para converter dados da mensagem de caractere de um conjunto de caracteres para outro... Para determinados conjuntos de caracteres de multibyte (por exemplo, conjuntos de caracteres UTF-16 ), as opções apropriadas devem ser usadas.


Nenhuma outra chamada MQI pode ser feita de dentro da saída; uma tentativa de fazer tal chamada falha com o código de razão MQRC\_CALL\_IN\_PROGRESS.

Consulte “MQXCNV-Converter caracteres” na página 952 para obter informações adicionais sobre a chamada MQXCNV e opções apropriadas.


## Utilitário para criação de código de saída de conversão


Use estas informações para saber mais sobre a criação do código de conversão de saída.

Os comandos para criar o código de saída de conversão são:

 **IBM i**  
CVTMQMDTA (Converter Tipo de Dados IBM MQ )

 **Sistemas AIX, Linux, and Windows**  
crtmqcvx (Criar IBM MQ conversão-saída).

 **z/OS**  
CSQUCVX

O comando para sua plataforma produz um fragmento de código que executa conversão de dados em estruturas de tipo de dados, para uso em seu programa de saída de conversão de dados. O comando usa um arquivo contendo uma ou mais definições de estrutura de linguagem C.  On z/OS, em seguida, ele gera um conjunto de dados que contém fragmentos de código assembler e funções de conversão. Em outras plataformas, ele gera um arquivo com uma função C para converter cada definição de estrutura.. No z/OS, o utilitário requer acesso à biblioteca de tempo de execução LE/370 SCEERUN..

## Chamando o utilitário CSQUCVX em z/OS



Figura 10 na página 1514 mostra um exemplo da JCL usada para chamar o utilitário CSQUCVX.

```
//CVX      EXEC PGM=CSQUCVX
//STEPLIB DD DISP=SHR,DSN=thlqua1.SCSQANLE
//         DD DISP=SHR,DSN=thlqua1.SCSQLOAD
//         DD DISP=SHR,DSN=1e370qua1.SCEERUN
//SYSPRINT DD SYSOUT=*
//CSQUINP DD DISP=SHR,DSN=MY.MQSERIES.FORMATS(MSG1)
//CSQUOUT DD DISP=OLD,DSN=MY.MQSERIES.EXIT(SMSG1)
```

Figura 10. JCL de amostra usado para chamar o utilitário CSQUCVX

## z/OS Instruções de definição de dados



O utilitário CSQUCVX requer instruções DD com os seguintes nomes DD mostrados em [Tabela 817](#) na página 1514:

Tabela 817. Nomes e descrições de instrução de definição de dados	
Instrução DD	Descrição
SYSPRINT	Especifica um conjunto de dados ou uma classe de spool de impressão para relatórios e mensagens de erros
CSQUINP	Especifica o conjunto de dados particionados que contém as definições das estruturas de dados a serem convertidas
CSQUOUT	Especifica o conjunto de dados particionados no qual os fragmentos de códigos de conversão devem ser gravados O comprimento de registro lógico (LRECL) deve ser 80 e o formato de registro (RECFM) deve ser FB.

## Mensagens de erro em sistemas AIX, Linux, and Windows .

O comando `crtmqcvx` retorna mensagens no intervalo AMQ7953 até AMQ7970.

Essas mensagens são lidas em [Mensagens e códigos de razão IBM MQ Mensagens](#)

Há dois tipos principais de erro:

- Erros principais, como erros de sintaxe, quando o processamento não pode continuar  
Uma mensagem é exibida na tela fornecendo o número da linha do erro no arquivo de entrada.. O arquivo de saída pode ter sido criado parcialmente
- Outros erros quando uma mensagem é exibida indicando que um problema foi localizado, mas que a análise da estrutura pode continuar.

O arquivo de saída foi criado e contém informações de erro sobre os problemas ocorridos. Essas informações de erro são prefixadas por `#ERROR` para que o código produzido não seja aceito por nenhum compilador sem intervenção para corrigir os problemas..

## Sintaxe válida

Seu arquivo de entrada para o utilitário deve estar em conformidade com a sintaxe da linguagem C..

Se não estiver familiarizado com C, consulte o [Exemplo C](#) neste tópico.

Além disso, esteja ciente das seguintes regras:

- typedef é reconhecido apenas antes da palavra-chave struct.
- Uma tag de estrutura é necessária em suas declarações de estrutura
- Você pode usar colchetes vazios [] para denotar uma matriz ou sequência de comprimento variável no final de uma mensagem.
- Matrizes multidimensionais e matrizes de sequências não são suportadas.
- Os seguintes tipos de dados adicionais são reconhecidos:
  - MQBOOL
  - MQBYTE
  - MQCHAR
  - MQFLOAT32
  - MQFLOAT64
  - MQSHORT
  - MQLONG
  - MQINT8
  - MQUINT8
  - MQINT16
  - MQUINT16
  - MQINT32
  - MQUINT32
  - MQINT64
  - MQUINT64

Os campos MQCHAR são convertidos para a página de código, mas MQBYTE, MQINT8 e MQUINT8 são deixados inalterados. Se a codificação for diferente, MQSHORT, MQLONG, MQINT16, MQUINT16, MQINT32, MQUINT32, MQINT64, MQUINT64, MQFLOAT32, MQFLOAT64 e MQBOOL serão convertidas adequadamente.

- Não use os seguintes tipos de dados:
  - double
  - Ponteiros
  - campos de bits

Isso ocorre porque o utilitário para criar código de saída de conversão não fornece o recurso para converter esses tipos de dados. Para superar isso, você pode escrever suas próprias rotinas e chamá-las da saída.

Outros pontos a observar:

- Não use números de sequência no conjunto de dados de entrada..
- Se houver campos para os quais deseja fornecer suas próprias rotinas de conversão, declare-os como MQBYTE e, em seguida, substitua as macros CMQXCFBA geradas por seu próprio código de conversão.

## Exemplo C

```
struct TEST { MQLONG    SERIAL_NUMBER;
              MQCHAR    ID[5];
              MQINT16   VERSION;
              MQBYTE    CODE[4];
              MQLONG    DIMENSIONS[3];
              MQCHAR    NAME[24];
            } ;
```

Isso corresponde às seguintes declarações em outras linguagens de programação:

## COBOL

```
10 TEST.  
15 SERIAL-NUMBER PIC S9(9) BINARY.  
15 ID PIC X(5).  
15 VERSION PIC S9(4) BINARY.  
* CODE IS NOT TO BE CONVERTED  
15 CODE PIC X(4).  
15 DIMENSIONS PIC S9(9) BINARY OCCURS 3 TIMES.  
15 NAME PIC X(24).
```

## System/390

```
TEST EQU *  
SERIAL_NUMBER DS F  
ID DS CL5  
VERSION DS H  
CODE DS XL4  
DIMENSIONS DS 3F  
NAME DS CL24
```

## PL/I

### Suportado em z/OS apenas

```
DCL 1 TEST,  
2 SERIAL_NUMBER FIXED BIN(31),  
2 ID CHAR(5),  
2 VERSION FIXED BIN(15),  
2 CODE CHAR(4), /* not to be converted */  
2 DIMENSIONS(3) FIXED BIN(31),  
2 NAME CHAR(24);
```

## MQ\_PUBLISH\_EXIT - saída Publish

A chamada MQ\_PUBLISH\_EXIT pode inspecionar e alterar mensagens entregues aos assinantes.

### Propósito

Use a saída de publicação para inspecionar e alterar mensagens entregues aos assinantes:

- Examine o conteúdo de uma mensagem publicada para cada assinante
- Modifique os conteúdos de uma mensagem publicada para cada assinante
- Altere a fila na qual uma mensagem é colocada
- Pare a entrega de uma mensagem para um assinante

Essa saída não está disponível em IBM MQ for z/OS

### Sintaxe

**MQ\_PUBLISH\_EXIT** (*ExitParms*, *PubContext*, *SubContext*)

### Parâmetros

#### ***ExitParms*** (MQPSXP) - Input/Output

*ExitParms* contém informações sobre a chamada da saída..

#### ***PubContext*** (MQPBC) - Input

*PubContext* contém informações contextuais sobre o editor da publicação.



## SubContext (MQSBC) - Input/Output

*SubContext* contém informações contextuais sobre o assinante recebendo a publicação.

## MQPSXP-Estrutura de dados de saída de publicação.

A estrutura MQPSXP descreve as informações que são transmitidas e retornadas da saída de publicação.

Tabela 818 na página 1517 resume os campos na estrutura:

Tabela 818. Campos em MQPSXP	
Campo	Descrição
<u>StrucID</u>	Identificador de estruturação
<u>Version</u>	Número de versão da estrutura
<u>ExitId</u>	Tipo de saída que está sendo chamado
<u>ExitReason</u>	Motivo para chamar a saída.
<u>ExitResponse</u>	Resposta da saída
<u>ExitResponse2</u>	Resposta secundária da saída
<u>Feedback</u>	Código de feedback
<u>ExitUserArea</u>	Sair da área do usuário..
<u>ExitData</u>	Dados de Saída
<u>QMgrName</u>	Nome do gerenciador de filas locais
<u>Hconn</u>	Identificador de conexões
<u>MsgDescPtr</u>	Endereço do descritor de mensagem (MQMD)
<u>MsgHandle</u>	Identificador para propriedades de mensagem (MQHMSG)
<u>MsgInPtr</u>	Endereço da mensagem de entrada
<u>MsgInLength</u>	Comprimento da mensagem de entrada
<u>MsgOutPtr</u>	Endereço da mensagem de saída
<u>MsgOutLength</u>	Comprimento da mensagem de saída:
<u>pEntryPoints</u>	Endereço da estrutura MQIEP

## Campos

### **StrucID (MQCHAR4)**

*StrucID* é o identificador de estrutura. O valor é o seguinte:

#### **MQPSXP\_STRUCID**

MQPSXP\_STRUCID é o identificador para a estrutura do parâmetro de saída de publicação Para a linguagem de programação C, a constante MQPSXP\_STRUC\_ID\_ARRAY também é definida; ela possui o mesmo valor que MQPSXP\_STRUC\_ID, mas é uma matriz de caracteres em vez de uma cadeia.

*StrucID* é um campo de entrada para a saída

### **Version (MQLONG)**

*Version* é o número da versão da estruturação O valor é o seguinte:

#### **MQPSXP\_VERSION\_1**

MQPSXP\_VERSION\_1 é a estrutura do parâmetro de saída de publicação da Versão 1 A constante MQPSXP\_CURRENT\_VERSION também é definida com o mesmo valor.

*Version* é um campo de entrada para a saída

### **ExitId (MQLONG)**

*ExitId* é o tipo de saída que está sendo chamado O valor é o seguinte:

#### **MQXT\_PUBLISH\_EXIT**

Saída de publicação.

*ExitId* é um campo de entrada para a saída

### **ExitReason (MQLONG)**

*ExitReason* é o motivo para chamar a saída Os valores possíveis são:

#### **MQXR\_INIT**

A saída para esta conexão é chamada para inicialização A saída pode adquirir e inicializar os recursos necessários; por exemplo, armazenamento principal.

#### **MQXR\_TERM**

A saída para esta conexão é chamada porque a saída está prestes a ser interrompida A saída deve liberar todos os recursos que ela adquiriu desde que foi inicializada; por exemplo, armazenamento principal

#### **MQXR\_PUBLICATION**

A saída é chamada pelo gerenciador de filas antes que ele coloque uma publicação em uma fila de mensagens de um assinante A saída pode alterar a mensagem, não colocar a mensagem na fila ou parar a publicação.

*ExitReason* é um campo de entrada para a saída

### **ExitResponse (MQLONG)**

Configure *ExitResponse* na saída para especificar como o processamento deve continuar

*ExitResponse* é um dos seguintes valores:

#### **MQXCC\_OK**

Configure MQXCC\_OK para continuar o processamento normalmente.. Configure MQXCC\_OK em resposta a quaisquer valores de *ExitReason*..

Se *ExitReason* tiver o valor MQXR\_PUBLICATION, os campos *DestinationQName* e *DestinationQMgrName* da estrutura MQSBC identificarão o destino para o qual a mensagem é enviada..

#### **MQXCC\_FAILED**

Configure MQXCC\_FAILED para parar a operação de publicação.. O código de conclusão MQCC\_FAILED e o código de razão 2557 (09FD) (RC2557): MQRC\_PUBLISH\_EXIT\_ERROR é configurado no retorno da saída.

#### **MQXCC\_SUPPRESS\_FUNCTION**

Configure MQXCC\_SUPPRESS\_FUNCTION para parar o processamento normal da mensagem.. Configure MQXCC\_SUPPRESS\_FUNCTION se *ExitReason* tiver o valor MQXR\_PUBLICATION.

A mensagem continua sendo processada pelo gerenciador de filas de acordo com a opção MQRO\_DISCARD\_MSG no campo *Report* no descritor de mensagens da mensagem.

- Se a opção MQRO\_DISCARD\_MSG for especificada a mensagem não será entregue ao assinante.
- Se a opção MQRO\_DISCARD\_MSG não for especificada, a mensagem será colocada na fila de mensagens não entregues Se não houver nenhuma fila de mensagens não entregues ou a mensagem não puder ser colocada com êxito na fila de mensagens não entregues, a publicação não será entregue ao assinante A entrega da publicação para outros assinantes depende dos valores dos atributos do objeto do tópico PMSGDLV e NPMSGDLV Para obter uma explicação desses atributos, consulte as descrições de parâmetros do comando DEFINE TOPIC .

*ExitResponse* é um campo de saída da saída.

### **ExitResponse2 (MQLONG)**

O *ExitResponse2* é reservado para uso futuro

### **Feedback (MQLONG)**

*Feedback* é o código de feedback a ser usado caso a saída retorne MQXCC\_SUPPRESS\_FUNCTION em *ExitResponse*

Na entrada para a saída, *Feedback* sempre tem o valor MQFB\_NONE. Se a saída retornar MQXCC\_SUPPRESS\_FUNCTION, configure *Feedback* para o valor a ser usado para a mensagem quando o gerenciador de filas o colocar na fila de mensagens não entregues. No retorno da saída, se *Feedback* tiver o valor original MQFB\_NONE, o gerenciador de filas configurará *Feedback* como MQFB\_STOPPED\_BY\_PUBSUB\_EXIT.

*Feedback* é um campo de entrada / saída para a saída..

### **ExitUserArea (MQBYTE16)**

*ExitUserArea* é um campo que está disponível para a saída usar. Cada conexão tem um *ExitUserArea* separado. O comprimento de *ExitUserArea* é fornecido por MQ\_EXIT\_USER\_AREA\_LENGTH

O campo *ExitReason* possui o valor MQXR\_INIT na primeira chamada da saída.. *ExitUserArea* é inicializado para MQXUA\_NONE na primeira invocação da saída para uma conexão As mudanças subsequentes para *ExitUserArea* são preservadas nas chamadas da saída

*ExitUserArea* é um campo de entrada / saída para a saída..

### **ExitData (MQCHAR32)**

*ExitData* é os dados de saída fixos definidos pelo parâmetro **PublishExitData** da sub-rotina no arquivo de inicialização do gerenciador de filas Os dados são preenchidos com espaços em branco até o comprimento total do campo Se não houver dados de saída fixos definidos no arquivo de inicialização, *ExitData* estará em branco.. O comprimento de *ExitData* é fornecido por MQ\_EXIT\_DATA\_LENGTH

*ExitData* é um campo de entrada para a saída

### **QMgrName (MQCHAR48)**

*QMgrName* é o nome do gerenciador de filas locais O nome é preenchido com espaços em branco até o comprimento total do campo O comprimento desse campo é fornecido por MQ\_Q\_MGR\_NAME\_LENGTH

*QMgrName* é um campo de entrada para a saída

### **Hconn (MQHCONN)**

*Hconn* é a manipulação representando uma conexão com o gerenciador de filas. Use *Hconn* apenas como um parâmetro para as chamadas de função de propriedade de mensagens MQSETMP, MQINQMMP ou MQDLTMP para trabalhar com propriedades de mensagens.

*Hconn* é um campo de entrada para a saída

### **MsgDescPtr (PMQMD)**

*MsgDescPtr* é o endereço do descritor de mensagens (MQMD) da mensagem sendo processada e é uma cópia do MQMD retornado da chamada MQPUT. A saída pode alterar o conteúdo do descritor de mensagem Qualquer mudança no conteúdo do descritor de mensagens deve ser feita com cuidado Especificamente, no caso em que o campo *SubType* da estrutura MQSBC é de valor MQSUBTYPE\_PROXY, o campo *CorrelId* no descritor de mensagens não deve ser alterado.

Nenhum descritor de mensagens é transmitido para a saída se *ExitReason* for MQXR\_INIT ou MQXR\_TERM ; nesses casos, *MsgDescPtr* é o ponteiro nulo.

*MsgDescPtr* é um campo de entrada para a saída

### **MsgHandle (MQHMSG)**

*MsgHandle* é o identificador para as propriedades de mensagens Use *MsgHandle* apenas com as chamadas de função de propriedades de mensagens MQSETMP, MQINQMMP ou MQDLTMP para trabalhar com propriedades de mensagens.

*MsgHandle* é um campo de entrada para a saída

### **MsgInPtr (PMQVOID)**

*MsgInPtr* é o endereço dos dados da mensagem de entrada. O conteúdo do buffer endereçado por *MsgInPtr* pode ter sido modificado pela saída; consulte [MsgOutPtr](#).

*MsgInPtr* é um campo de entrada para a saída.

### **MsgInLength (MQLONG)**

*MsgInLength* é o comprimento em bytes dos dados da mensagem transmitidos para a saída. O endereço dos dados é fornecido por *MsgInPtr*.

*MsgInLength* é um campo de entrada para a saída.

### **MsgOutPtr (PMQVOID)**

*MsgOutPtr* é o endereço de um buffer que contém dados da mensagem que são retornados da saída. Na entrada para a saída, *MsgOutPtr* é nulo. No retorno da saída, se o valor ainda for nulo, o gerenciador de filas enviará a mensagem especificada por *MsgInPtr*, com o comprimento fornecido por *MsgInLength*.

Se a saída modificar os dados da mensagem, use um dos seguintes procedimentos:

- Se o comprimento dos dados não for alterado, os dados poderão ser modificados no buffer endereçado por *MsgInPtr*. Nesse caso, não mude *MsgOutPtr* e *MsgOutLength*.
- Se os dados modificados forem menores que os dados originais, os dados poderão ser modificados no buffer endereçado por *MsgInPtr*. Neste caso, *MsgOutPtr* deve ser configurado para o endereço do buffer de mensagem de entrada, e *MsgOutLength* configurado para o novo comprimento dos dados da mensagem.
- Se os dados modificados forem, ou puderem ser, maiores que os dados originais, a saída deverá obter um novo buffer de mensagem. Copie os dados modificados para ele. Configure *MsgOutPtr* para o endereço do novo buffer e configure *MsgOutLength* para o comprimento dos novos dados da mensagem. A saída é responsável por liberar o buffer endereçado por *MsgOutPtr* quando a saída é chamada pela próxima vez.

**Nota:** *MsgOutPtr* é sempre o ponteiro nulo na entrada para a saída e não o endereço de um buffer de mensagem anteriormente obtido. Para liberar o buffer obtido anteriormente a saída deve salvar seu endereço e comprimento. Salve as informações em *ExitUserArea*, ou em um bloco de controle que tenha seu endereço salvo em *ExitUserArea*.

*MsgOutPtr* é um campo de entrada / saída para a saída..

### **MsgOutLength (MQLONG)**

*MsgOutLength* é o comprimento em bytes dos dados da mensagem retornados pela saída. Na entrada da saída, este campo é sempre zero. No retorno da saída, esse campo será ignorado se *MsgOutPtr* for nulo. Consulte [MsgOutPtr](#) para obter informações sobre como modificar os dados da mensagem..

*MsgOutLength* é um campo de entrada / saída para a saída..

### **pEntryPoints (PMQIEP)**

*pEntryPoints* é o endereço de uma estrutura MQIEP por meio da qual as chamadas MQI e DCI podem ser feitas.

## **Declaração de linguagem C-MQPSXP.**

```
typedef struct tagMQPSXP {
    MQCHAR4   StructId;           /* Structure identifier */
    MQLONG    Version;           /* Structure version number */
    MQLONG    ExitId;            /* Type of exit */
    MQLONG    ExitReason;        /* Reason for invoking exit */
    MQLONG    ExitResponse;      /* Response from exit */
    MQLONG    ExitResponse2;     /* Reserved */
    MQLONG    Feedback;         /* Feedback code */
    MQBYTE16  ExitUserArea;      /* Exit user area */
    MQCHAR32  ExitData;          /* Exit data */
    MQCHAR48  QMgrName;         /* Name of local queue manager */
    MQHCONN   Hconn;            /* Connection handle */
}
```

```

MQHMSG      MsgHandle;          /* Handle to message properties */
MQMD        MsgDescPtr;        /* Address of message descriptor */
MQVOID      MsgInPtr;         /* Address of input message data */
MQLONG      MsgInLength;      /* Length of input message data */
MQVOID      MsgOutPtr;        /* Address of output message data */
MQLONG      MsgOutLength;     /* Length of output message data */
/* Ver:1 */
MQIEP       pEntryPoints;     /* Address of the MQIEP structure */
/* Ver:2 */
} MQPSXP;

```

## MQPBC-Estrutura de dados do contexto de publicação

A estrutura MQPBC contém as informações contextuais, relacionadas ao publicador da publicação, que são passadas para a saída de publicação

Tabela 819 na página 1521 resume os campos na estrutura:

Tabela 819. Campos em MQPBC	
Campo	Descrição
<i>StrucID</i>	Identificador de estruturação
<i>Version</i>	Número de versão da estrutura
<i>PubTopicString</i>	Sequência de tópicos de publicação
<i>MsgDescPtr</i>	Endereço do descritor de mensagem (MQMD)

### Campos

#### **StrucID (MQCHAR4)**

*StrucID* é o identificador de estrutura. O valor é o seguinte:

##### **MQPBC\_STRUCID**

MQPBC\_STRUCID é o identificador para a estrutura de contexto de publicação Para a linguagem de programação C, a constante MQPBC\_STRUC\_ID\_ARRAY também é definida; ela possui o mesmo valor que MQPBC\_STRUC\_ID, mas é uma matriz de caracteres em vez de uma cadeia.

*StrucID* é um campo de entrada para a saída

#### **Version (MQLONG)**

*Version* é o número da versão da estruturação O valor é o seguinte:

##### **MQPBC\_VERSION\_1**

MQPBC\_VERSION\_1 é a estrutura do parâmetro de saída de publicação da Versão 1

##### **MQPBC\_VERSION\_2**

MQPBC\_VERSION\_2 é a estrutura do parâmetro de saída de publicação Versão 2. A constante MQPBC\_CURRENT\_VERSION também é definida com o mesmo valor.

*Version* é um campo de entrada para a saída

#### **PubTopicString (MQCHARV)**

*PubTopicString* é a sequência de tópicos que está sendo publicada.

*PubTopicString* é um campo de entrada para a saída

#### **MsgDescPtr (MQMD)**

*MsgDescPtr* é o endereço de uma cópia do descritor de mensagens (MQMD) para a mensagem sendo processada.

*MsgDescPtr* é um campo de entrada para a saída

## Declaração de linguagem C-MQPBC.

```
typedef struct tagMQPBC {
```

```

MQCHAR4   StrucId;           /* Structure identifier */
MQLONG    Version;          /* Structure version number */
MQCHARV   PubTopicString;   /* Publish topic string */
PMQMD     MsgDescPtr;       /* Address of message descriptor */
} MQPBC;

```

## MQSBC-Estrutura de dados do contexto de assinatura

A estrutura MQSBC contém as informações contextuais, relacionadas ao assinante que está recebendo a publicação, que são passadas para a saída de publicação

Tabela 820 na página 1522 resume os campos na estrutura:

Tabela 820. Campos em MQSBC	
Campo	Descrição
<u>StrucID</u>	Identificador de estruturação
<u>Version</u>	Número de versão da estrutura
<u>DestinationQMGrName</u>	Nome do gerenciador de filas de destino
<u>DestinationQName</u>	Nome da fila de destino
<u>SubType</u>	Tipo de assinatura
<u>SubOptions</u>	Opções de Assinatura
<u>ObjectName</u>	Nome do objeto
<u>ObjectString</u>	Sequência do objeto
<u>SubTopicString</u>	Sequência de tópicos de assinatura
<u>SubName</u>	Nome da assinatura
<u>SubId</u>	Identificador de Assinatura
<u>SelectionString</u>	Endereço da sequência de seleção
<u>SubLevel</u>	Nível da assinatura
<u>PSPProperties</u>	Propriedades de Publicação / Assinatura

### Campos

#### **StrucID (MQCHAR4)**

Identificador de estruturação. O valor é o seguinte:

##### **MQSBC\_STRUCID**

MQSBC\_STRUCID é o identificador para a estrutura do parâmetro de saída de publicação Para a linguagem de programação C, a constantes MQSBC\_STRUC\_ID\_ARRAY também é definida; MQSBC\_STRUC\_ID\_ARRAY tem o mesmo valor que MQSBC\_STRUC\_ID, mas é uma matriz de caracteres em vez de uma cadeia.

*StrucID* é um campo de entrada para a saída

#### **Version (MQLONG)**

Número de versão da estrutura. O valor é o seguinte:

##### **MQSBC\_VERSION\_1**

Estrutura do parâmetro de saída de publicação da versão 1 A constante MQSBC\_CURRENT\_VERSION também é definida com o mesmo valor.

*Version* é um campo de entrada para a saída

**DestinationQMgrName (MQCHAR48)**

*DestinationQMgrName* é o nome do gerenciador de filas ao qual a mensagem está sendo enviada. O nome é preenchido com espaços em branco até o comprimento total do campo O nome pode ser alterado pela saída. O comprimento desse campo é fornecido por MQ\_Q\_MGR\_NAME\_LENGTH

*DestinationQMgrName* é um campo de entrada / saída para a saída; consulte a [nota](#)

**DestinationQName (MQCHAR48)**

*DestinationQName* é o nome da fila para a qual a mensagem está sendo enviada O nome é preenchido com espaços em branco até o comprimento total do campo O nome pode ser alterado pela saída. O comprimento desse campo é fornecido por MQ\_Q\_NAME\_LENGTH

*DestinationQName* é um campo de entrada / saída para a saída; consulte a [nota](#)

**SubType (MQLONG)**

*SubType* indica como a assinatura foi criada. Os valores válidos são MQSUBTYPE\_API, MQSUBTYPE\_ADMIN e MQSUBTYPE\_PROXY ; consulte [Consultar Status da Assinatura \(Resposta\)](#).

*SubType* é um campo de entrada para a saída

**SubOptions (MQLONG)**

*SubOptions* são as opções de subscrição; consulte [“Opções \(MQLONG\) para MQSD” na página 593](#) para obter uma descrição dos valores que este campo pode utilizar

*SubOptions* é um campo de entrada para a saída

**ObjectName (MQCHAR48)**

*ObjectName* é o nome do objeto do tópico, conforme definido no gerenciador de filas locais O comprimento desse campo é fornecido por MQ\_TOPIC\_NAME\_LENGTH O nome do objeto é o nome do objeto do tópico administrativo que o gerenciador de filas associou à sequência de tópicos Mesmo se o assinante forneceu um objeto de tópico como parte da assinatura, o *ObjectName* pode ser um objeto de tópico diferente.. A associação de um objeto do tópico com uma subscrição depende da resolução integral de *SubTopicString*

*ObjectName* é um campo de entrada para a saída

**ObjectString (MQCHARV)**

*ObjectString* é a sequência de tópicos completa da publicação que foi inscrita. Todos os curingas na cadeia de assinatura original são resolvidos É diferente do campo MQSD subscription *ObjectString* descrito em [“ObjectString \(MQCHARV\) para MQSD” na página 603](#), que pode conter curingas e é exclusivo de qualquer nome de objeto fornecido pelo assinante.

*ObjectString* é um campo de entrada para a saída

**SubTopicString (MQCHARV)**

*SubTopicString* é a sequência de tópicos completa fornecida pelo assinante. *SubTopicString* é a combinação da sequência de tópicos definida em um objeto do tópico e de uma sequência de tópicos Um assinante deve fornecer um objeto de tópico, uma sequência de tópicos, ou ambos Se o assinante fornecer uma sequência de tópicos, ele poderá conter curingas

*SubTopicString* é um campo de entrada para a saída

**SubName (MQCHARV)**

*SubName* é o nome da assinatura fornecido pelo assinante ou é um nome gerado.

*SubName* é um campo de entrada para a saída

**SubId (MQBYTE 24)**

*SubId* é o identificador de assinatura interno exclusivo.

*SubId* é um campo de entrada para a saída

**SelectionString (MQCHARV)**

*SelectionString* é o critério de seleção usado ao assinar mensagens de um tópico; consulte [Seletores](#).

*SelectionString* é um campo de entrada para a saída

### SubLevel (MQLONG)

*SubLevel* é o nível de interceptação associado à assinatura; consulte [“SubLevel \(MQLONG\) para MQSD”](#) na página 607 para obter detalhes adicionais

*SubLevel* é um campo de entrada para a saída

### PSProperties (MQLONG)

*PSProperties* são as propriedades de publicação / assinatura. Eles especificam como as propriedades de mensagem relacionadas à publicação / assinatura são incluídas nas mensagens enviadas para essa assinatura. Os valores possíveis são MQPSPROP\_NONE, MQPSPROP\_COMPAT, MQPSPROP\_RFH2, MQPSPROP\_MSGPROP. Consulte [Parâmetros opcionais \(Alterar, Copiar e Criar Assinatura\)](#) para obter uma descrição desses valores.

*PSProperties* é um campo de entrada para a saída

**Nota:** As verificações de autorização são executadas apenas nos valores originais de *DestinationQMgrName* e *DestinationQName* antes de serem transmitidas para a saída de publicação.. Nenhuma nova verificação de autorização será executada quando a saída alterar a fila de destino, alterando *DestinationQMgrName* ou *DestinationQName*

## Declaração de idioma C-MQSB

```
typedef struct tagMQSB {
    MQCHAR4   StructId;           /* Structure identifier */
    MQLONG    Version;           /* Structure version number */
    MQCHAR48  DestinationQMgrName; /* Destination queue manager */
    MQCHAR48  DestinationQName;  /* Destination queue name */
    MQLONG    SubType;           /* Type of subscription */
    MQLONG    SubOptions;        /* Subscription options */
    MQCHAR48  ObjectName;        /* Object name */
    MQCHARV   ObjectString;      /* Object string */
    MQCHARV   SubTopicString;    /* Subscription topic string */
    MQCHARV   SubName;           /* Subscription name */
    MQBYTE24  SubId;             /* Subscription identifier */
    MQCHARV   SelectionString;    /* Subscription selection string */
    MQLONG    SubLevel;          /* Subscription level */
    MQLONG    PSProperties;      /* Publish/subscribe properties */
} MQSB;
```

## Chamadas de Saída do Canal e Estrutura de Dados

Essa coleção de tópicos fornece informações de referência sobre as chamadas especiais do IBM MQ e estruturas de dados que podem ser usadas ao gravar programas de saída do canal.

Essas informações são informações da interface de programação sensível ao produto. É possível gravar saídas de usuário do IBM MQ nas seguintes linguagens de programação:

Tabela 821. IBM MQ saídas de usuário: plataformas e linguagens de programação	
Plataforma	Linguagens de Programação
IBM MQ for z/OS	Assembler e C (que devem estar em conformidade com o ambiente de programação do sistema C para saídas do sistema, descrito no <i>z/OS C/C++ Programming Guide</i> .)
IBM MQ for IBM i	ILE C, ILE COBOL e ILE RPG
Todas as outras plataformas IBM MQ	C

Também é possível gravar saídas de usuário no Java para uso apenas com aplicativos Java e JMS. Para obter mais informações sobre como criar e usar saídas de canal com o IBM MQ classes for Java, consulte [Usando saídas de canal em IBM MQ classes for Java](#) e para IBM MQ classes for JMS, consulte [Usando saídas de canal com IBM MQ classes for JMS](#).



Não é possível gravar saídas de usuário IBM MQ no TAL ou Visual Basic. No entanto, uma declaração para a estrutura MQCD é fornecida no Visual Basic para uso na chamada MQCONNX de um programa IBM MQ MQI client .

Em vários casos nas descrições a seguir, os parâmetros são matrizes ou sequências de caracteres com um tamanho que não é corrigido. Para esses parâmetros, um "n" minúsculo é usado para representar uma constante numérica. Quando a declaração para esse parâmetro for codificada, o "n" deverá ser substituído pelo valor numérico necessário. Para obter informações adicionais sobre as convenções usadas nessas descrições, consulte o [“Tipos de dados elementares”](#) na página 237

## arquivos de definição de dados

Arquivos de definição de dados são fornecidos com IBM MQ para cada uma das linguagens de programação suportadas. Para obter detalhes desses arquivos, consulte [Copiar, cabeçalho, inclusão e arquivos de módulo](#).

### MQ\_CHANNEL\_EXIT-Saída de canal.

A chamada MQ\_CHANNEL\_EXIT descreve os parâmetros que são transmitidos para cada uma das saídas de canal chamadas pelo Agente do canal de mensagens...

Nenhum ponto de entrada chamado MQ\_CHANNEL\_EXIT é fornecido pelo gerenciador de filas; o nome MQ\_CHANNEL\_EXIT não tem significado especial, pois os nomes das saídas de canal são fornecidos no MQCD de definição de canal.

Há cinco tipos de saída de canal:

- Saída de segurança do canal
- Saída de mensagem do canal
- Saída de envio do canal
- Saída de recebimento do canal
- Mensagem do canal-saída de nova tentativa

Os parâmetros são semelhantes para cada tipo de saída e a descrição fornecida aqui se aplica a todos eles, exceto onde especificamente indicado.

## Sintaxe

**MQ\_CHANNEL\_EXIT** (*ChannelExitParms*, *ChannelDefinition*, *DataLength*,  
*AgentBufferLength*, *AgentBuffer*, *ExitBufferLength*, *ExitBufferAddr*)

## Parâmetros

A chamada MQ\_CHANNEL\_EXIT possui os seguintes parâmetros.

### ChannelExitParâmetros (MQCXP)-entrada/saída

Bloco do parâmetro de saída do canal

Essa estrutura contém informações adicionais relacionadas à invocação da saída. A saída configura as informações nessa estrutura para indicar como o MCA continua.

### ChannelDefinition (MQCD)-entrada/saída

Definição de canal.

Essa estrutura contém os parâmetros configurados pelo administrador para controlar o comportamento do canal.

### DataLength (MQLONG)-entrada/saída

Comprimento dos dados..

Os dados dependem do tipo de saída:

- Para uma saída de segurança do canal, quando a saída é chamada, esse parâmetro contém o comprimento de qualquer mensagem de segurança no campo *AgentBuffer*, se *ExitReason* for MQXR\_SEC\_MSG. Será zero se não houver mensagem. A saída deve configurar esse campo para o comprimento de qualquer mensagem de segurança a ser enviada para seu parceiro se ele configurar *ExitResponse* para MQXCC\_SEND\_SEC\_MSG ou MQXCC\_SEND\_AND\_REQUEST\_SEC\_MSG. Os dados da mensagem estão em *AgentBuffer* ou *ExitBufferAddr*.

O conteúdo das mensagens de segurança é da exclusiva responsabilidade das saídas.

- Para uma saída de mensagem do canal, quando a saída for chamada, esse parâmetro conterá o comprimento da mensagem (incluindo o cabeçalho da fila de transmissão). A saída deve configurar esse campo para o comprimento da mensagem em *AgentBuffer* ou *ExitBufferAddr* que deve continuar. Isso deve ser maior ou igual ao comprimento do cabeçalho da fila de transmissão (MQXQH).
- Para uma saída de envio de canal ou de recepção de canal, quando a saída for chamada, esse parâmetro conterá o comprimento da transmissão. A saída deve configurar esse campo para o comprimento da transmissão em *AgentBuffer* ou *ExitBufferAddr* que deve continuar.

Se uma saída de segurança enviar uma mensagem e não houver saída de segurança na outra extremidade do canal ou a outra extremidade configurar um *ExitResponse* de MQXCC\_OK, a saída inicial será chamada novamente com MQXR\_SEC\_MSG e uma resposta nula (*DataLength* = 0).

### **AgentBufferComprimento (MQLONG)-entrada**

Comprimento do buffer do agente..

Esse parâmetro pode ser maior que *DataLength* na invocação.

Para mensagens do canal, saídas de envio e recebimento, qualquer espaço não utilizado na chamada pode ser usado pela saída para expandir os dados no local. Se isso for feito, o parâmetro **DataLength** deverá ser configurado adequadamente pela saída.

Na linguagem de programação C, esse parâmetro é transmitido pelo endereço.

### **AgentBuffer (Comprimento de MQBYTE x AgentBuffer)-entrada/saída**

Buffer do agente..

O conteúdo desse parâmetro depende do tipo de saída:

- Para uma saída de segurança do canal, na chamada da saída ele contém uma mensagem de segurança se *ExitReason* for MQXR\_SEC\_MSG. Para enviar uma mensagem de segurança de volta, a saída pode usar esse buffer ou seu próprio buffer (*ExitBufferAddr*).
- Para uma saída de mensagem do canal, na chamada da saída este parâmetro contém:

- O cabeçalho da fila de transmissão (MQXQH), que inclui o descritor de mensagens (que em si contém as informações de contexto para a mensagem), imediatamente seguido por
- Os dados da mensagem

Se a mensagem for continuar, a saída poderá executar um dos seguintes procedimentos:

- Deixe o conteúdo do buffer inalterado
- Modificar o conteúdo no local (retornando o novo comprimento dos dados em *DataLength*; não deve ser maior que *AgentBufferLength*)
- Copie o conteúdo para o *ExitBufferAddr*, fazendo quaisquer mudanças necessárias

Quaisquer mudanças que a saída faz no cabeçalho da fila de transmissão não são verificadas; no entanto, modificações erradas podem significar que a mensagem não pode ser colocada no destino.

- Para uma saída de envio ou de recebimento de canal, na chamada da saída, isso contém os dados de transmissão. A saída pode fazer um dos seguintes:
  - Deixe o conteúdo do buffer inalterado

- Modificar o conteúdo no local (retornando o novo comprimento dos dados em *DataLength* ; não deve ser maior que *AgentBufferLength*)
  - Copie o conteúdo para o *ExitBufferAddr*, fazendo quaisquer mudanças necessárias
- Os primeiros 8 bytes dos dados não devem ter sido alterados pela saída.

### ExitBufferComprimento (MQLONG)-entrada/saída

O comprimento do buffer de saída.

Na primeira chamada da saída, esse parâmetro é configurado como zero. Depois disso, qualquer valor que for transmitido de volta pela saída, em cada chamada, será apresentado à saída na próxima vez que ela for chamada.. O valor não é usado pelo MCA.

**Nota:** Esse parâmetro não deve ser usado por saídas gravadas em linguagens de programação que não suportem o tipo de dados do ponteiro

### ExitBufferAddr (MQPTR)-entrada/saída

Endereço do buffer de saída..

Esse parâmetro é um ponteiro para o endereço de um buffer de armazenamento gerenciado pela saída, no qual ele pode optar por retornar dados de mensagem ou transmissão (dependendo do tipo de saída) para o agente se o buffer do agente for ou não for grande o suficiente ou se for mais conveniente para a saída fazer isso.

Na primeira chamada da saída, o endereço transmitido à saída é nulo. Depois disso, qualquer endereço transmitido de volta pela saída, em cada chamada, será apresentado à saída na próxima vez em que for chamado.

Se *ExitBufferAddr* for nulo, os dados usados serão obtidos do parâmetro *AgentBuffer* .

Se *ExitBufferAddr* não for nulo, os dados utilizados serão obtidos do buffer apontado pelo parâmetro *ExitBufferAddr*.

**Nota:** Este parâmetro não deve ser usado por saídas gravadas em linguagens de programação que não suportem o tipo de dados do ponteiro

## Chamada C

```
exitname (&ChannelExitParms, &ChannelDefinition,
&DataLength, &AgentBufferLength, AgentBuffer,
&ExitBufferLength, &ExitBufferAddr);
```

Os parâmetros transmitidos para a saída são declarados da seguinte forma:

```
MQCXP ChannelExitParms; /* Channel exit parameter block */
MQCD ChannelDefinition; /* Channel definition */
MQLONG DataLength; /* Length of data */
MQLONG AgentBufferLength; /* Length of agent buffer */
MQBYTE AgentBuffer[n]; /* Agent buffer */
MQLONG ExitBufferLength; /* Length of exit buffer */
MQPTR ExitBufferAddr; /* Address of exit buffer */
```

## Chamada COBOL

```
CALL 'exitname' USING CHANNELEXITPARMS, CHANNELDEFINITION,
DATALENGTH, AGENTBUFFERLENGTH, AGENTBUFFER,
EXITBUFFERLENGTH, EXITBUFFERADDR.
```

Os parâmetros transmitidos para a saída são declarados da seguinte forma:

```
** Channel exit parameter block
01 CHANNELEXITPARMS.
```

```

COPY CMQCPV.
** Channel definition
01 CHANNELDEFINITION.
COPY CMQCDV.
** Length of data
01 DATALENGTH PIC S9(9) BINARY.
** Length of agent buffer
01 AGENTBUFFERLENGTH PIC S9(9) BINARY.
** Agent buffer
01 AGENTBUFFER PIC X(n).
** Length of exit buffer
01 EXITBUFFERLENGTH PIC S9(9) BINARY.
** Address of exit buffer
01 EXITBUFFERADDR POINTER.

```

## Chamada de RPG (ILE)

```

C*.1.....2.....3.....4.....5.....6.....7..
C          CALLP          exitname(MQCP : MQCD : DATLEN :
C                               ABUFL : ABUF : EBUFL :
C                               EBUF)

```

A definição de protótipo para a chamada é:

```

D*.1.....2.....3.....4.....5.....6.....7..
Dexitname PR          EXTPROC('exitname')
D* Channel exit parameter block
D MQCP 160A
D* Channel definition
D MQCD 1328A
D* Length of data
D DATLEN 10I 0
D* Length of agent buffer
D ABUFL 10I 0
D* Agent buffer
D ABUF * VALUE
D* Length of exit buffer
D EBUFL 10I 0
D* Address of exit buffer
D EBUF *

```

## Chamada do assembler System/390

```

CALL EXITNAME, (CHANNELEXITPARMS, CHANNELDEFINITION, DATALENGTH, X
AGENTBUFFERLENGTH, AGENTBUFFER, EXITBUFFERLENGTH, X
EXITBUFFERADDR)

```

Os parâmetros transmitidos para a saída são declarados da seguinte forma:

CHANNELEXITPARMS	CMQCPA	,	Channel exit parameter block
CHANNELDEFINITION	CMQCDA	,	Channel definition
DATALENGTH	DS	F	Length of data
AGENTBUFFERLENGTH	DS	F	Length of agent buffer
AGENTBUFFER	DS	CL(n)	Agent buffer
EXITBUFFERLENGTH	DS	F	Length of exit buffer
EXITBUFFERADDR	DS	F	Address of exit buffer

## Observações de Uso

1. A função executada pela saída do canal é definida pelo provedor da entrada. A saída, entretanto, deve estar em conformidade com as regras definidas aqui e no bloco de controle associado, o MQCPX.
2. O parâmetro **ChannelDefinition** transmitido para a saída do canal pode ser uma das várias versões.. Consulte o campo *Version* na estrutura MQCD para obter mais informações..

3. Se a saída do canal receber uma estrutura MQCD com o campo *Version* configurado para um valor maior que MQCD\_VERSION\_1, a saída deverá usar o campo *ConnectionName* em MQCD, em preferência ao campo *ShortConnectionName* ..
4. Em geral, as saídas do canal podem alterar o comprimento dos dados da mensagem. Isso pode surgir como resultado da saída incluir dados na mensagem ou remover dados da mensagem ou compactar ou criptografar a mensagem. No entanto, restrições especiais se aplicarão se a mensagem for um segmento que contenha apenas parte de uma mensagem lógica. Em particular, não deve haver mudança líquida no comprimento da mensagem como resultado das ações de saídas complementares de envio e recebimento..

Por exemplo, é permitido que uma saída de envio reduza a mensagem compactando-a, mas a saída de recebimento complementar deve restaurar o comprimento original da mensagem descompactando-a, para que não haja mudança líquida no comprimento da mensagem.

Essa restrição surge porque alterar o comprimento de um segmento faria com que os deslocamentos de segmentos posteriores na mensagem fossem incorretos e isso inibiria a capacidade do gerenciador de filas de reconhecer que os segmentos formaram uma mensagem lógica completa.

## MQ\_CHANNEL\_AUTO\_DEF\_EXIT-Saída de definição automática do canal

A chamada MQ\_CHANNEL\_AUTO\_DEF\_EXIT descreve os parâmetros que são passados para a saída de autodefinição de canal chamada pelo Agente do canal de mensagens...

Nenhum ponto de entrada chamado MQ\_CHANNEL\_AUTO\_DEF\_EXIT é fornecido pela gerenciador de filas; o nome MQ\_CHANNEL\_AUTO\_DEF\_EXIT não é de significado especial porque os nomes das saídas de definição automática são fornecidos no gerenciador de filas

### Sintaxe

**MQ\_CHANNEL\_AUTO\_DEF\_EXIT (*ChannelExitParms*, *ChannelDefinition*)**

### Parâmetros

A chamada MQ\_CHANNEL\_AUTO\_DEF\_EXIT tem os seguintes parâmetros.

#### ChannelExitParâmetros (MQCXP)-entrada/saída

Bloco do parâmetro de saída do canal

Essa estrutura contém informações adicionais relacionadas à invocação da saída. A saída configura as informações nessa estrutura para indicar como o MCA continua.

#### ChannelDefinition (MQCD)-entrada/saída

Definição de canal.

Esta estrutura contém parâmetros configurados pelo administrador para controlar o comportamento dos canais que são criados automaticamente. A saída configura informações nessa estrutura para modificar o comportamento padrão configurado pelo administrador.

Os campos MQCD listados não devem ser alterados pela saída:

- *ChannelName*
- *ChannelType*
- *StrucLength*
- *Version*

Se outros campos forem alterados, o valor configurado pela saída deverá ser válido. Se o valor não for válido, uma mensagem de erro será gravada no arquivo do log de erro ou exibida no console (conforme apropriado para o ambiente)...



**Atenção:** Os canais autodefinidos criados por uma saída de channel automatic definition (CHAD) não podem configurar o rótulo certificado, porque o handshake TLS ocorreu pelo tempo que o

canal é criado. Configurar o rótulo certificado em uma saída CHAD para canais de entrada não terá efeito.

## Chamada C

```
exitname (&ChannelExitParms, &ChannelDefinition);
```

Os parâmetros transmitidos para a saída são declarados da seguinte forma:

```
MQCXP ChannelExitParms; /* Channel exit parameter block */
MQCD ChannelDefinition; /* Channel definition */
```

## Chamada COBOL

```
CALL 'exitname' USING CHANNELEXITPARMS, CHANNELDEFINITION.
```

Os parâmetros transmitidos para a saída são declarados da seguinte forma:

```
** Channel exit parameter block
01 CHANNELEXITPARMS.
   COPY CMQXCPV.
** Channel definition
01 CHANNELDEFINITION.
   COPY CMQCDV.
```

## Chamada de RPG (ILE)

```
C*.1.....2.....3.....4.....5.....6.....7..
C          CALLP          exitname(MQCXP : MQCD)
```

A definição de protótipo para a chamada é:

```
D*.1.....2.....3.....4.....5.....6.....7..
Dexitname PR          EXTPROC('exitname')
D* Channel exit parameter block
D MQCXP          160A
D* Channel definition
D MQCD          1328A
```

## Chamada do assembler System/390

```
CALL EXITNAME, (CHANNELEXITPARMS, CHANNELDEFINITION)
```

Os parâmetros transmitidos para a saída são declarados da seguinte forma:

```
CHANNELEXITPARMS CMQXCPA , Channel exit parameter block
CHANNELDEFINITION CMQCDA , Channel definition
```

## Observações de Uso

1. A função executada pela saída do canal é definida pelo provedor da entrada. A saída, entretanto, deve estar em conformidade com as regras definidas aqui e no bloco de controle associado, o MQCXP.
2. O parâmetro **ChannelExitParms** transmitido para a saída de definição automática do canal é uma estrutura MQCXP.. A versão de MQCXP transmitida depende do ambiente no qual a saída está em


execução; consulte a descrição do campo *Version* em [“MQCXP-Parâmetro de saída do canal”](#) na página 1574 para obter detalhes..

3. O parâmetro **ChannelDefinition** transmitido para a saída de definição automática do canal é uma estrutura MQCD.. A versão do MQCD transmitida depende do ambiente no qual a saída está em execução; consulte a descrição do campo *Version* em [“MQCD-Definição de canal”](#) na página 1532 para obter detalhes..

## MQXWAIT-Espera na saída

A chamada MQXWAIT aguarda um evento ocorrer. Ele pode ser usado apenas a partir de uma saída de canal em z/OS...

O uso de MQXWAIT ajuda a evitar problemas de desempenho que podem, de outra forma, ocorrer se uma saída de canal fizer algo que cause uma espera. O evento MQXWAIT está aguardando é sinalizado por um MVS ECB (bloco de controle de eventos). O BCE é descrito na descrição do bloco de controle do MQXWD

 Para obter mais informações sobre o uso de MQXWAIT e gravar programas de saída do canal, consulte [Gravando programas de saída do canal em z/OS](#)

### Sintaxe

**MQXWAIT** (*Hconn*, *WaitDesc*, *CompCode*, *Reason*)

### Parâmetros

A chamada MQXWAIT possui os seguintes parâmetros.

#### Hconn (MQHCONN)-entrada

Manipulação de conexões.

Esta manipulação representa a conexão ao gerenciador de filas. O valor de *Hconn* foi retornado por uma chamada MQCONN anterior emitida na chamada da mesma ou anterior da saída..

#### WaitDesc (MQXWD)-entrada/saída.

Descritor de espera.

Este parâmetro descreve o evento pelo qual aguardar.. Consulte [“MQXWD-Descritor de espera de saída”](#) na página 1589 para obter detalhes dos campos nesta estrutura..

#### CompCode (MQLONG)-saída

Código de conclusão.

É um dos seguintes códigos:

##### MQCC\_OK

Indica conclusão bem-sucedida.

##### MQCC\_FAILED

A chamada falhou.

#### Motivo (MQLONG)-saída

Código de razão qualificando *CompCode*.

Se *CompCode* for MQCC\_OK:

##### MQRC\_NONE

(0, X'000') Nenhuma razão a relatar.

##### MQRC\_ADAPTER\_NOT\_AVAILABLE

(2204, X'89C') Adaptador não disponível.

##### MQRC\_OPTIONS\_ERROR

(2046, X'7FE') Opções não válidas ou não consistentes.

## **MQRC\_XWAIT\_CANCELED**

(2107, X'83B') chamada MQXWAIT cancelada.

## **MQRC\_XWAIT\_ERROR**

(2108, X'83C') A chamada de chamada MQXWAIT não é válida.

## **Chamada C**

```
MQXWAIT (Hconn, &WaitDesc, &CompCode, &Reason);
```

Declare os parâmetros conforme a seguir:

```
MQHCONN  Hconn;      /* Connection handle */
MQXWD    WaitDesc;  /* Wait descriptor */
MQLONG   CompCode; /* Completion code */
MQLONG   Reason;   /* Reason code qualifying CompCode */
```

## **Chamada do assembler System/390**

```
CALL MQXWAIT,(HCONN,WAITDESC,COMPCODE,REASON)
```

Declare os parâmetros conforme a seguir:

```
HCONN      DS          F  Connection handle
WAITDESC   CMQXWDA    ,  Wait descriptor
COMPCODE   DS          F  Completion code
REASON     DS          F  Reason code qualifying COMPCODE
```

## **MQCD-Definição de canal**

A estrutura MQCD contém parâmetros que controlam a execução de um canal. Ele é transmitido para cada saída de canal que é chamada de um Agente do Canal de Mensagens (MCA).

Para obter mais informações sobre saídas do canal, consulte [“MQ\\_CHANNEL\\_EXIT-Saída de canal.”](#) na [página 1525](#). A descrição neste tópico está relacionada aos canais de mensagens e aos canais MQI.

## **Campos de nome de saída..**

Quando uma saída é chamada, o campo relevante de *SecurityExit*, *MsgExit*, *SendExit*, *ReceiveExit* e *MsgRetryExit* contém o nome da saída atualmente sendo chamada. O significado do nome nesses campos depende do ambiente no qual o MCA está em execução.. Exceto onde indicado, o nome é alinhado à esquerda dentro do campo, sem espaços em branco integrados; o nome é preenchido com espaços em branco para o comprimento do campo. Nas descrições a seguir, os colchetes ([]) denotam informações opcionais:

### **AIX and Linux**

O nome da saída é o nome de um módulo ou biblioteca carregável dinamicamente, sufixado com o nome de uma função residente nessa biblioteca. O nome da função deve estar entre parênteses. Opcionalmente, o nome da biblioteca pode ser prefixado com um caminho do diretório:

```
[ path ] library ( function )
```

O nome é limitado a no máximo 128 caracteres.

### **z/OS**

O nome da saída é o nome de um módulo de carregamento que é válido para especificação no parâmetro EP da macro LINK ou LOAD.. O nome é limitado a no máximo oito caracteres.



## Windows

O nome da saída é o nome de uma biblioteca de link dinâmico, sufixado com o nome de uma função residente nessa biblioteca. O nome da função deve estar entre parênteses. O nome da biblioteca pode, opcionalmente, ser prefixado com um caminho de diretório e unidade:

```
[d:][ path ] library ( function )
```

O nome é limitado a no máximo 128 caracteres.

## IBM i

O nome da saída é um nome de programa de 10 bytes seguido por um nome de biblioteca de 10 bytes. Se os nomes tiverem menos de 10 bytes, cada nome será preenchido com espaços em branco para torná-lo 10 bytes. O nome da biblioteca pode ser \*LIBL , exceto ao chamar uma saída de definição automática de canal, nesse caso um nome completo é necessário.

## Mudando campos MQCD em uma saída de canal

Uma saída do canal pode mudar os campos no MQCD. O valor alterado permanece no MQCD e é transmitido para quaisquer saídas restantes em uma cadeia de saída e para qualquer conversa que compartilhe a instância do canal. O MQCD alterado também é usado pelo MCA para seu processamento normal durante o tempo de vida do canal.

Os seguintes campos MQCD não devem ser alterados pela saída:

- ChannelName
- ChannelType
- StrucLength
- Versão

## Referências relacionadas

[“Campos” na página 1533](#)

Este tópico lista todos os campos na estrutura MQCD e descreve cada campo..

[“Declaração C” na página 1561](#)

Esta é a declaração C para a estrutura MQCD.

[“Declaração COBOL” na página 1563](#)

Esta declaração é a declaração de COBOL para a estrutura MQCD

[“Declaração RPG \(ILE\)” na página 1565](#)

Esta é a declaração RPG para a estrutura MQCD.

[“Declaração do assembler System/390” na página 1568](#)

Essa declaração é a declaração do assembler System/390 para a estrutura MQCD..

[“Declaração do Visual Basic” na página 1569](#)

Esta declaração é a declaração Visual Basic da estrutura MQCD.

[“Mudando campos MQCD em uma saída de canal” na página 1571](#)

Uma saída do canal pode mudar os campos no MQCD. No entanto, essas mudanças geralmente não são realizadas, exceto nas circunstâncias listadas.

## Campos

Este tópico lista todos os campos na estrutura MQCD e descreve cada campo..

*BatchDataLimit (MQLONG)*

Este campo especifica o limite, em kilobytes, da quantidade de dados que podem ser enviados por um canal antes de obter um ponto de sincronização.

Um ponto de sincronização é tomado depois que a mensagem que fez com que o limite fosse atingido fluiu através do canal.

O lote é finalizado quando uma das seguintes condições é satisfeita:

- Mensagens **BatchSize** foram enviadas.
- **BatchDataLimit** bytes foram enviados.
- A fila de transmissão está vazia e **BatchInterval** foi excedido..

O valor deve estar no intervalo de 0 a 999999. O valor-padrão é 5000.

Um valor zero neste atributo significa que nenhum limite de dados é aplicado a lotes sobre este canal.

Esse parâmetro aplica-se apenas a canais com um *ChannelType* de MQCHT\_SENDER, MQCHT\_SERVER, MQCHT\_CLUSRCVR ou MQCHT\_CLUSSDR

Este é um campo de entrada para a saída O campo não estará presente se *Version* for menor que MQCD\_VERSION\_11..

#### *BatchHeartbeat (MQLONG)*

Esse campo especifica o intervalo de tempo usado para acionar uma pulsação em lote para o canal.

A pulsação em lote permite que os canais emissores determinem se a instância do canal remoto ainda está ativa antes de ficar pendente. Uma pulsação em lote ocorre se um canal emissor não tiver se comunicado com a instância do canal remoto dentro do intervalo de tempo especificado.

O valor está no intervalo de 0 a 999 999; as unidades são milissegundos. Um valor zero indica que a pulsação em lote não está ativada.

Esse campo é relevante apenas para canais que possuem um *ChannelType* de MQCHT\_SENDER, MQCHT\_SERVER, MQCHT\_CLUSSDR, ou MQCHT\_CLUSRCVR

Este é um campo de entrada para a saída O campo não estará presente se *Version* for menor que MQCD\_VERSION\_7.

#### *BatchInterval (MQLONG)*

Este campo especifica o tempo aproximado em milissegundos que um canal mantém um lote aberto, se menos de *BatchSize* mensagens tiverem sido transmitidas no lote atual.

Se *BatchInterval* for maior que zero, o lote será finalizado por qualquer um dos eventos a seguir que ocorrer primeiro:

- Mensagens *BatchSize* foram enviadas ou
- *BatchInterval* milissegundos decorreram desde o início do lote.

Se *BatchInterval* for zero, o lote será finalizado por qualquer um dos eventos a seguir que ocorrer primeiro:

- Mensagens *BatchSize* foram enviadas ou
- a fila de transmissão fica vazia.

*BatchInterval* deve estar no intervalo de zero a 999 999 999.

Esse campo é relevante apenas para canais com um *ChannelType* de MQCHT\_SENDER, MQCHT\_SERVER, MQCHT\_CLUSSDR, ou MQCHT\_CLUSRCVR

Este é um campo de entrada para a saída O campo não está presente quando o *Version* é menor que MQCD\_VERSION\_4

#### *BatchSize (MQLONG)*

Este campo especifica o número máximo de mensagens que podem ser enviadas através de um canal antes de sincronizar o canal.

Esse campo não é relevante para canais com um *ChannelType* de MQCHT\_SVRCON ou MQCHT\_CLNTCONN.

#### *CertificateLabel (MQCHAR64)*

Este campo fornece detalhes do rótulo do certificado que está sendo usado

IBM MQ inicializa o valor padrão para o campo *CertificateLabel* como espaços em branco.

Isto é interpretado no tempo de execução como o valor padrão e é compatível com versões anteriores.

Por exemplo, especificar uma versão do MQCD menor que 11 ou usar o valor padrão de espaços em branco para o campo *CertificateLabel* significa que esse campo é ignorado.

O comprimento desse campo é fornecido por MQ\_CERT\_LABEL\_LENGTH

#### *ChannelMonitoring (MQLONG)*

Este campo especifica o nível atual de coleta de dados de monitoramento do canal.

Esse campo não é relevante para canais com um ChannelType de MQCHT\_CLNT\_CONN.

Ele é um dos seguintes valores:

- MQMON\_OFF
- MQMON\_LOW
- MQMON\_MEDIUM
- MQMON\_HIGH

Este é um campo de entrada para a saída. Ele não estará presente se *Version* for menor que MQCD\_VERSION\_8..

#### *ChannelName (MQCHAR20)*

Este campo especifica o nome da definição de canal.

Deve haver uma definição de canal com o mesmo nome na máquina remota para poder se comunicar.

O nome deve usar apenas os caracteres:

- Maiúsculos de A a Z
- Minúsculos de a a z
- Numéricos de 0 a 9
- Ponto (.)
- Barra (/)
- Sublinhado (\_)
- Sinal de percentual (%)

e ser preenchido à direita com espaços em branco. Os espaços em branco integrado ou iniciais não são permitidos.

O comprimento desse campo é fornecido por MQ\_CHANNEL\_NAME\_LENGTH

#### *ChannelStatistics (MQLONG)*

Este campo especifica o nível atual de coleta de dados estatísticos para o canal..

Este campo não é relevante para canais com um ChannelType de MQCHT\_CLNT\_CONN ou MQCHT\_SVRCONN.

Ele é um dos seguintes valores:

- MQMON\_OFF
- MQMON\_LOW
- MQMON\_MEDIUM
- MQMON\_HIGH

Este é um campo de entrada para a saída. Ele não estará presente se *Version* for menor que MQCD\_VERSION\_8..

#### *ChannelType (MQLONG)*

Este campo especifica o tipo de canal.

Ele é um dos seguintes valores:

**MQCHT\_SENDER**

Emissor.

**MQCHT\_SERVER**

Servidor.

**MQCHT\_RECEIVER**

Receptor.

**MQCHT\_REQUESTER**

Solicitante.

**MQCHT\_CLNTCONN**

Conexão do cliente..

**MQCHT\_SVRCONN**

Conexão do servidor (para uso pelos clientes).

**MQCHT\_CLUSSDR**

Emissor de cluster..

**MQCHT\_CLUSRCVR**

Receptor do cluster

*ClientChannelPeso (MQLONG)*

Esse campo especifica um peso para influenciar qual definição de canal de conexão do cliente é utilizada...

O atributo *Peso* de *ClientChannel* é usado para que as definições de canais do cliente possam ser selecionadas aleatoriamente com base em seu peso quando mais de uma definição adequada estiver disponível.. Quando um cliente emite uma *MQCONN* solicitando conexão com um grupo de gerenciadores de filas, especificando um nome de gerenciador de filas começando com um asterisco e mais de uma definição de canal adequada está disponível na tabela de definição de canal do cliente (CCDT), a definição a ser usada é selecionada aleatoriamente com base na ponderação, com quaisquer definições de *Peso* do *ClientChannel* aplicáveis selecionadas primeiro em ordem alfabética.

Especifique um valor no intervalo de 0 - 99. O padrão é 0.

Um valor 0 indica que nenhum balanceamento de carga é desempenhado e as definições aplicáveis são selecionadas em ordem alfabética. Para ativar o balanceamento de carga, escolha um valor no intervalo de 1 a 99, em que 1 é o peso mais baixo e 99 é o mais alto. A distribuição de mensagens entre dois ou mais canais com pesos diferentes de zero é proporcional à proporção desses pesos. Por exemplo, três canais com valores de peso *ClientChannel* de 2, 4 e 14 são selecionados aproximadamente 10%, 20% e 70% do tempo. Esta distribuição não é garantida

Esse atributo é válido apenas para o tipo de canal de conexão do cliente..

Este é um campo de entrada para a saída O campo não estará presente se *Versão* for menor que *MQCD\_VERSION\_9*.

*ClusterPtr (MQPTR)*

Este campo especifica o endereço de uma lista de nomes de clusters..

Se *ClustersDefined* for maior que zero, esse endereço será o endereço de uma lista de nomes de clusters.. O canal pertence a cada cluster listado.

Esse campo é relevante apenas para canais com um *ChannelType* de *MQCHT\_CLUSSDR* ou *MQCHT\_CLUSRCVR*..

Este é um campo de entrada para a saída O campo não estará presente se *Version* for menor que *MQCD\_VERSION\_5*.

*ClustersDefined (MQLONG)*

Este campo especifica o número de clusters aos quais o canal pertence..

Esse campo é o número de nomes de clusters apontado por *ClusterPtr* É zero ou maior.

Esse campo é relevante apenas para canais com um *ChannelType* de MQCHT\_CLUSSDR ou MQCHT\_CLUSRCVR..

Este é um campo de entrada para a saída O campo não estará presente se *Version* for menor que MQCD\_VERSION\_5.

#### *CLWLChannelPriority (MQLONG)*

Este campo especifica a prioridade do canal de carga de trabalho do cluster

O algoritmo de escolha do gerenciador de carga de trabalho seleciona um destino com a prioridade mais alta do conjunto de destinos selecionados com base na classificação. Se houver dois possíveis gerenciadores de fila de destino, esse atributo poderá ser usado para fazer um failover do gerenciador de fila no outro gerenciador de filas Todas as mensagens vão para o gerenciador de filas com a prioridade mais alta até que isso termine, em seguida, as mensagens vão para o gerenciador de filas com a próxima prioridade mais alta.

O valor está no intervalo de 0 a 9. O padrão é 0.

Este é um campo de entrada para a saída O campo não estará presente se *Version* for menor que MQCD\_VERSION\_8.

Para obter informações adicionais, consulte [Configurando um Cluster do Gerenciador de Filas](#)

#### *CLWLChannelRank (MQLONG)*

Este campo especifica a classificação do canal de carga de trabalho do cluster

O algoritmo de escolha do gerenciador de carga de trabalho seleciona um destino com a classificação mais alta. Quando o destino final é um gerenciador de filas em um cluster diferente, é possível configurar a classificação de gerenciadores de fila de gateway intermediários (na interseção de clusters vizinhos) para que o algoritmo de escolha escolha corretamente um gerenciador de fila de destino mais próximo do destino final.

O valor está no intervalo de 0 a 9. O padrão é 0.

Este é um campo de entrada para a saída O campo não estará presente se *Version* for menor que MQCD\_VERSION\_8.

Para obter informações adicionais, consulte [Configurando um Cluster do Gerenciador de Filas](#)

#### *CLWLChannelWeight (MQLONG)*

Este campo especifica o peso do canal de carga de trabalho do cluster.

Peso do canal de carga de trabalho do cluster

O algoritmo de escolha do gerenciador de carga de trabalho usa o atributo "weight" do canal para distorcer a opção de destino para que mais mensagens possam ser enviadas para uma máquina específica. Por exemplo, é possível fornecer a um canal em um servidor UNIX grande um "peso" maior do que outro canal no PC desktop pequeno e o algoritmo de escolha escolhe o servidor UNIX com mais frequência do que o PC..

O valor está no intervalo de 1 a 99 O padrão é 50.

Este é um campo de entrada para a saída O campo não estará presente se *Version* for menor que MQCD\_VERSION\_8.

Para obter informações adicionais, consulte [Configurando um Cluster do Gerenciador de Filas](#)

#### *ConnectionAffinity (MQLONG)*

Esse campo especifica se os aplicativos clientes que se conectam várias vezes usando o mesmo nome do gerenciador de filas usam o mesmo canal do cliente.

Utilize esse atributo quando várias definições de canal aplicáveis estiverem disponíveis.

O valor é um dos seguintes:

## **MQCAFTY\_PREFERRED**

A primeira conexão em um processo que lê uma tabela de definições de canais do cliente (CCDT) cria uma lista de definições aplicáveis com base no peso com quaisquer definições CLNTWGHT (0) aplicáveis primeiro e em ordem alfabética. Cada conexão no processo tenta conectar-se utilizando a primeira definição na lista. Se uma conexão for malsucedida, a próxima definição será utilizada. Definições malsucedidas com valores CLNTWGHT diferentes de 0 são movidas para o final da lista. As definições CLNTWGHT(0) permanecem no início da lista e são selecionadas primeiro para cada conexão.

Cada processo de cliente com o mesmo nome de host sempre cria a mesma lista.

Para aplicativos clientes gravados em C, C++ ou na estrutura de programação .NET (incluindo totalmente gerenciado .NET), a lista será atualizada se a CCDT tiver sido modificada desde que a lista foi criada.

Esse valor é o valor padrão.

## **MQCAFTY\_NONE**

A primeira conexão em um processo que está lendo um CCDT cria uma lista de definições aplicáveis. Todas as conexões em um processo selecionam uma definição aplicável baseada em peso com quaisquer definições de CLNTWGHT(0) aplicáveis selecionadas primeiro em ordem alfabética.

Para aplicativos clientes gravados em C, C++ ou na estrutura de programação .NET (incluindo totalmente gerenciado .NET), a lista será atualizada se a CCDT tiver sido modificada desde que a lista foi criada.

Esse atributo é válido apenas para o tipo de canal de conexão do cliente..

Este é um campo de entrada para a saída O campo não estará presente se *Versão* for menor que MQCD\_VERSION\_9.

### *ConnectionName (MQCHAR264)*

Este campo especifica o nome de conexão do canal.

Para canais receptores de cluster (quando especificado) CONNAME relaciona-se ao gerenciador de filas local, e para outros canais, ao gerenciador de filas de destino. O valor especificado depende do protocolo de transmissão (*TransportType*) a ser usado:

- Para MQXPT\_LU62, é o nome completo da Unidade Lógica parceira.
- Para MQXPT\_NETBIOS, é o nome NetBIOS definido na máquina remota.
- Para MQXPT\_TCP, ele é o nome do host, o endereço de rede da máquina remota especificado em IPv4 decimal pontilhado ou IPv6 formato hexadecimal ou a máquina local para canais do receptor de clusters.
- Para MQXPT\_SPX, é um endereço de estilo SPX que compreende um endereço de rede de 4 bytes, um endereço de nó de 6 bytes e um número de soquete de 2 bytes.

Ao definir um canal, esse campo não é relevante para canais com um *ChannelType* de MQCHT\_SVRCONN ou MQCHT\_RECEIVER. No entanto, quando a definição de canal é transmitida para uma saída, esse campo contém o endereço do parceiro, independentemente do tipo de canal.

O comprimento desse campo é fornecido por MQ\_CONN\_NAME\_LENGTH Esse campo não estará presente se *Version* for menor que MQCD\_VERSION\_2.

### *DataConversion (MQLONG)*

Esse campo especifica se o agente do canal de mensagem de envio tenta a conversão dos dados da mensagem do aplicativo se o agente do canal de mensagens de recebimento não puder executar essa conversão.

Esse campo se aplica somente a mensagens que não são segmentos de mensagens lógicas; o MCA nunca tenta converter mensagens que são segmentos.

Esse campo é relevante apenas para canais com um *ChannelType* de MQCHT\_SENDER, MQCHT\_SERVER, MQCHT\_CLUSSDR, ou MQCHT\_CLUSRCVR É um dos seguintes:

## **MQDC\_SENDER\_CONVERSION**

Conversão por emissor.

## **MQDC\_NO\_SENDER\_CONVERSION**

Nenhuma conversão pelo remetente

### *DefReconnect ( MQLONG)*

O atributo do canal DefReconnect configura o valor do atributo de reconexão padrão para um canal de conexão do cliente..

A opção de reconexão do cliente automática padrão. É possível configurar um IBM MQ MQI client para que reconecte automaticamente um aplicativo cliente. O IBM MQ MQI client tenta se reconectar a um gerenciador de filas após uma falha de conexão. Ele tenta se reconectar sem o aplicativo cliente emitir uma chamada de MQI de MQCONN ou MQCONNX.

Reconexão é uma opção MQCONNX . Usando o atributo do canal DefReconnect , é possível incluir o comportamento de reconexão em aplicativos existentes que usam MQCONN. Também é possível mudar o comportamento da reconexão de aplicativos que usam o MQCONNX

Também é possível configurar o valor DefRecon do arquivo mqclient.ini para configurar ou modificar o comportamento de reconexão. O valor DefRecon do arquivo mqclient.ini tem precedência sobre o atributo do canal DefReconnect .

## **Syntax**

**DefReconnect** ( MQRCN\_NO (default) |MQRCN\_YES|MQRCN\_Q\_MGR|MQRCN\_DISABLED )

## **Parâmetros**

### **MQRCN\_NO**

MQRCN\_NO é o valor padrão.

A menos que substituído por **MQCONNX**, o cliente não é reconectado automaticamente.

### **MQRCN\_YES**

A menos que substituído por **MQCONNX**, o cliente se reconecta automaticamente.

### **MQRCN\_Q\_MGR**

A menos que substituído por **MQCONNX**, o cliente se reconecta automaticamente, mas apenas para o mesmo gerenciador de fila A opção QMGR tem o mesmo efeito que MQCNO\_RECONNECT\_Q\_MGR.

### **MQRCN\_DISABLED**

A reconexão é desativada, mesmo se solicitada pelo programa cliente usando a chamada MQI **MQCONNX** .

A reconexão de cliente automática não é suportada pelo IBM MQ classes for Java.

<b>DefReconnect</b>	<b>Opções de reconexão configuradas no aplicativo</b>			
	MQCNO_RECONNE CT	MQCNO_RECONNE CT_Q_MGR	MQCNO_RECONNE CT_AS_DEF	MQCNO_RECONNE CT_DISABLED
MQRCN_NO	SIM	QMGR	NÃO	NÃO
MQRCN_YES	SIM	QMGR	SIM	NÃO
MQRCN_Q_MGR	SIM	QMGR	QMGR	NÃO
MQRCN_DISABLED	NÃO	NÃO	NÃO	NÃO

## Conceitos relacionados

[Reconexão automática do cliente](#)

[Reconexão de Canal e Cliente](#)

[Sub-rotina CHANNELS do Arquivo de Configuração do Cliente](#)

## Referências relacionadas

[“Opções \(MQLONG\) para MQCNO” na página 329](#)

[Opções que controlam a ação de MQCONNX.](#)

*Desc (MQCHAR64)*

Esse campo pode ser usado para comentário descritivo

O conteúdo do campo não é significativo para os Agentes do Canal de Mensagens. No entanto, ele deve conter apenas caracteres que podem ser exibidos.. Ele não pode conter nenhum caractere nulo; se necessário, será preenchido à direita com espaços em branco. Em uma instalação DBCS, o campo pode conter caracteres DBCS (sujeito a um comprimento máximo de campo de 64 bytes).

**Nota:** Se esse campo contiver caracteres que não estejam no conjunto de caracteres do gerenciador de fila (conforme definido pelo atributo do gerenciador de filas **CodedCharSetId**) esses caracteres poderão ser convertidos incorretamente se esse campo for enviado para outro gerenciador de filas.

O comprimento desse campo é fornecido por MQ\_CHANNEL\_DESC\_LENGTH

*DiscInterval (MQLONG)*

Este campo especifica o tempo máximo em segundos durante o qual o canal aguarda uma mensagem chegar na fila de transmissão, antes de finalizar o canal.

Em outras palavras, especifica o intervalo de desconexão.

O valor A de zero faz com que o MCA aguarde indefinidamente.

Para canais de conexão do servidor usando o protocolo TCP, o intervalo representa o valor de desconexão de inatividade do cliente, especificado em segundos. Se uma conexão do servidor não tiver recebido nenhuma comunicação de seu cliente parceiro por esse período, ela finalizará a conexão. O intervalo de inatividade de conexão do servidor se aplica apenas entre chamadas API do IBM MQ de um cliente, portanto, nenhum cliente é desconectado durante uma chamada MQGET de longa execução com espera.

Esse atributo não é aplicável para canais de conexão do servidor que usam protocolos diferentes de TCP

Esse campo é relevante apenas para canais com um *ChannelType* de MQCHT\_SENDER, MQCHT\_SERVER, MQCHT\_CLUSSDR, MQCHT\_CLUSRCVR, ou MQCHT\_SVRCONN

*Comprimento de ExitData(MQLONG)*

Esse campo especifica o comprimento em bytes de cada um dos itens de dados do usuário nas listas de itens de dados do usuário de saída endereçados pelos campos *MsgUserDataPtr*, *SendUserDataPtr* e *ReceiveUserDataPtr*

Esse comprimento não é necessariamente o mesmo que MQ\_EXIT\_DATA\_LENGTH

Este é um campo de entrada para a saída. O campo não estará presente se *Version* for menor que MQCD\_VERSION\_4..

*Comprimento de ExitName(MQLONG)*

Esse campo especifica o comprimento em bytes de cada um dos nomes nas listas de nomes de saída endereçados pelos campos *MsgExitPtr*, *SendExitPtr* e *ReceiveExitPtr*.

Esse comprimento não é necessariamente o mesmo que MQ\_EXIT\_NAME\_LENGTH

Este é um campo de entrada para a saída. O campo não estará presente se *Version* for menor que MQCD\_VERSION\_4..

*HdrCompLista [2] (MQLONG)*

Este campo especifica a lista de técnicas de compactação de dados de cabeçalho suportadas pelo canal.

A lista contém um ou mais dos seguintes valores:



### **MQCOMPRESS\_NONE**

Nenhuma compactação de dados do cabeçalho é executada.

### **MQCOMPRESS\_SYSTEM**

A compactação de dados de cabeçalho é executada.

### **MQCOMPRESS\_NOT\_AVAILABLE**

Os valores não utilizados na lista são configurados para esse valor.

Este é um campo de entrada para a saída O campo não estará presente se *Version* for menor que MQCD\_VERSION\_8.

#### *Intervalo de Pulsação (MQLONG)*

Esse campo especifica o tempo em segundos entre os fluxos de pulsação.

A interpretação desse campo depende do tipo de canal, conforme a seguir:

- Para um tipo de canal de MQCHT\_SENDER, MQCHT\_SERVER, MQCHT\_RECEIVER MQCHT\_REQUESTER, MQCHT\_CLUSSDR ou MQCHT\_CLUSRCVR, esse campo é o tempo em segundos entre fluxos de pulsação transmitidos do MCA de envio quando não há mensagens na fila de transmissão. Isso dá ao MCA de recebimento a oportunidade de fazer quiesce do canal. Para ser útil, *HeartbeatInterval* deve ser menor que *DiscInterval*..
- Para um tipo de canal de MQCHT\_CLNTCONN ou MQCHT\_SVRCONN com o campo Conversas de Compartilhamento MQCD configurado como zero, esse campo é o tempo em segundos entre os fluxos de pulsação transmitidos do servidor MCA quando esse MCA emitiu uma chamada MQGET com a opção MQGMO\_WAIT em nome de um aplicativo cliente. Isso permite que o servidor MCA manipule situações em que a conexão do cliente falha durante um MQGET com MQGMO\_WAIT.
- Para um tipo de canal de MQCHT\_CLNTCONN ou MQCHT\_SVRCONN com o campo Conversas de Compartilhamento do MQCD configurado para um valor diferente de zero, esse campo é o tempo em segundos entre o fluxo de pulsação quando não há fluxos de dados enviados ou recebidos Isso permite que o canal seja colocado em modo quiesce de forma eficiente.

O valor está no intervalo de 0 a 999 999.. O valor que é usado é o maior dos valores especificados no lado de envio e no lado de recebimento, a menos que um valor 0 seja especificado em qualquer lado, nesse caso, nenhuma troca de pulsação ocorrerá.

Este é um campo de entrada para a saída O campo não estará presente se *Version* for menor que MQCD\_VERSION\_4..

#### *Intervalo KeepAlive(MQLONG)*

Este campo especifica o valor transmitido para a pilha de comunicações para a sincronização keepalive para o canal.

O valor é aplicável para os protocolos de comunicação TCP/IP e SPX, embora nem todas as implementações suportem esse parâmetro.

O valor está no intervalo de 0 a 99 999; as unidades são segundos. Um valor zero indica que o keep-alive do canal não está ativado, embora o keep-alive ainda possa ocorrer se o keep-alive do TCP/IP (em vez do keep-alive do canal) estiver ativado. O seguinte valor especial também é válido:

### **MQKAI\_AUTO**

Automático.

Esse valor indica que o intervalo keepalive é calculado a partir do intervalo de pulsação negociado, conforme a seguir:

- Se o intervalo de pulsação negociado for maior que zero, o intervalo de keep-alive usado será o intervalo de pulsação mais 60 segundos.
  - Se o intervalo de pulsação negociado for zero, o intervalo de keep-alive usado será zero.
- No z/OS, o keepalive TCP/IP ocorre quando TCPKEEP (YES) é especificado no objeto do gerenciador de filas.

- Em outros ambientes, o keep-alive do TCP/IP ocorre quando o parâmetro **KEEPALIVE=YES** é especificado na sub-rotina TCP no arquivo de configuração de enfileiramento distribuído

Esse campo é relevante apenas para canais que possuem um *TransportType* de MQXPT\_TCP ou MQXPT\_SPX.

Este é um campo de entrada para a saída O campo não estará presente se *Version* for menor que MQCD\_VERSION\_7.

#### *LocalAddress (MQCHAR48)*

Este campo especifica o endereço TCP/IP local definido para o canal para as comunicações de saída.

Este campo estará em branco se nenhum endereço específico for definido para comunicações de saída O endereço pode opcionalmente incluir um número de porta ou intervalo de números de portas. O formato deste endereço é:

```
[ip-addr][(low-port[,high-port])]
```

em que colchetes ([]) denotam informações opcionais, ip-addr é especificado em IPv4 decimal pontilhado, IPv6 hexadecimal ou alfanumérico e low-port e high-port são números de porta entre parênteses. Todos são opcionais.

Um endereço IP específico, porta ou intervalo de portas para comunicações de saída é útil em cenários de recuperação em que um canal é reiniciado em uma pilha TCP/IP diferente.

*LocalAddress* é semelhante em forma a *ConnectionName*, mas não deve ser confundido com ele *LocalAddress* especifica as características das comunicações locais, enquanto *ConnectionName* especifica como alcançar um gerenciador de filas remotas.

No IBM MQ 9.3.0, o Java Message Queueing Interface (JMQUI) foi atualizado para assegurar que o campo de endereço local seja configurado em um objeto MQCD após uma instância do canal ter sido criada e conectada a um gerenciador de filas. Isso significa que, quando uma saída de canal gravada em Java chama o método MQCD.getLocalAddress(), o método retorna o endereço local que a instância do canal está usando. Antes de IBM MQ 9.3.0, a saída de segurança do canal não pôde acessar o endereço local que está sendo usado pela instância do canal e o método MQCD.getLocalAddress() retornou nulo.

Esse campo é relevante apenas para canais com um *TransportType* de MQXPT\_TCP e um *ChannelType* de MQCHT\_SENDER, MQCHT\_SERVER, MQCHT\_REQUESTER, MQCHT\_CLNTCONN, MQCHT\_CLUSSDR ou MQCHT\_CLUSRCVR..

O comprimento desse campo é fornecido por MQ\_LOCAL\_ADDRESS\_LENGTH Esse campo não estará presente se o *Version* for menor que MQCD\_VERSION\_7

#### *LongMCAUserIdLength (MQLONG)*

Esse campo especifica o comprimento em bytes do identificador de usuário do MCA completo apontado por *LongMCAUserIdPtr*.

Esse campo não é relevante para os canais com um *ChannelType* de MQCHT\_CLNTCONN

Este é um campo de entrada / saída na saída. O campo não estará presente se o *Version* for menor que MQCD\_VERSION\_6

#### *LongMCAUserIdPtr (MQPTR)*

Este campo especifica o endereço do identificador de usuários do MCA longo..

Se *LongMCAUserIdLength* for maior que zero, este campo será o endereço do identificador de usuário do MCA completo... O comprimento do identificador completo é fornecido por *LongMCAUserIdLength* Os primeiros 12 bytes do identificador de usuário MCA também estão contidos no campo *MCAUserIdentifier*.

Consulte a descrição do campo *MCAUserIdentifier* para obter detalhes do identificador do usuário MCA.

Esse campo não é relevante para canais com um *ChannelType* de MQCHT\_SDR, MQCHT\_SV, MQCHT\_CLNTCONN ou MQCHT\_CLUSSDR.

Este é um campo de entrada / saída na saída. O campo não estará presente se o *Version* for menor que MQCD\_VERSION\_6

*LongRemoteUserIdComprimento (MQLONG)*

Este campo especifica o comprimento em bytes do identificador de usuário remoto completo apontado por *LongRemoteUserIdPtr*.

Esse campo é relevante apenas para canais com um *ChannelType* de MQCHT\_CLNTCONN ou MQCHT\_SVRCONN

Este é um campo de entrada para a saída O campo não estará presente se o *Version* for menor que MQCD\_VERSION\_6

*LongRemoteUserIdPtr (MQPTR)*

Este campo especifica o endereço do identificador de usuário remoto longo..

Se *LongRemoteUserIdLength* for maior que zero, essa sinalização será o endereço do identificador de usuário remoto completo. O comprimento do identificador completo é fornecido por *LongRemoteUserIdLength* Os primeiros 12 bytes do identificador de usuário remoto também estão contidos no campo *RemoteUserIdentifier*

Consulte a descrição do campo *RemoteUserIdentifier* para obter detalhes do identificador de usuário remoto.

Esse campo é relevante apenas para canais com um *ChannelType* de MQCHT\_CLNTCONN ou MQCHT\_SVRCONN

Este é um campo de entrada para a saída O campo não estará presente se o *Version* for menor que MQCD\_VERSION\_6

*LongRetryContagem (MQLONG)*

Esse campo especifica a contagem usada após a contagem especificada pelo *ShortRetryCount* ter sido esgotada

Ele especifica o número máximo de tentativas adicionais feitas para se conectar à máquina remota, em intervalos especificados por *LongRetryInterval*, antes de registrar um erro no operador.

Esse campo é relevante apenas para canais com um *ChannelType* de MQCHT\_SENDER, MQCHT\_SERVER, MQCHT\_CLUSSDR, ou MQCHT\_CLUSRCVR

*Intervalo de LongRetry(MQLONG)*

Este campo especifica o número máximo de segundos a aguardar, antes de tentar novamente a conexão com a máquina remota

O intervalo entre novas tentativas pode ser estendido se o canal precisar aguardar para se tornar ativo.

Esse campo é relevante apenas para canais com um *ChannelType* de MQCHT\_SENDER, MQCHT\_SERVER, MQCHT\_CLUSSDR, ou MQCHT\_CLUSRCVR

*MaxInstances (MQLONG)*

Este campo especifica o número máximo de instâncias simultâneas de um canal de conexão do servidor individual que podem ser iniciadas..

Este campo é usado apenas nos canais de conexão do servidor

O campo pode ter um valor no intervalo de 0 a 999 999 999. Um valor igual a zero impede o acesso de todos os clientes.

O valor padrão desse campo é 999 999 999.

Se o valor desse campo for reduzido para um número menor que o número de instâncias do canal de conexão do servidor que estão em execução atualmente, essas instâncias em execução não serão afetadas. No entanto, novas instâncias não podem ser iniciadas até que instâncias existentes suficientes tenham cessado a execução para que o número de instâncias atualmente em execução seja menor que o valor do campo.

#### *MaxInstancesPerClient (MQLONG)*

Este campo especifica o número máximo de instâncias simultâneas de um canal de conexão do servidor individual que pode ser iniciado a partir de um único cliente...

Neste contexto, as conexões originadas do mesmo endereço de rede remota são consideradas como provenientes do mesmo cliente.

Este campo é usado apenas nos canais de conexão do servidor

O campo pode ter um valor no intervalo de 0 a 999 999 999. Um valor igual a zero impede o acesso de todos os clientes.

O valor padrão desse campo é 999 999 999.

Se o valor desse campo for reduzido para um número menor que o número de instâncias do canal de conexão do servidor atualmente em execução a partir de clientes individuais, essas instâncias em execução não serão afetadas. No entanto, novas instâncias de qualquer um desses clientes não podem ser iniciadas até que instâncias existentes suficientes tenham cessado a execução de modo que o número de instâncias atualmente em execução, originadas do cliente que está tentando iniciar uma nova, seja menor que o valor do campo.

#### *Comprimento de MaxMsg(MQLONG)*

Este campo especifica o comprimento máximo da mensagem que pode ser transmitida no canal..

Ele é comparado com o valor para o canal remoto, e o máximo real é o menor dos dois valores.

#### *MCAName (MQCHAR20)*

Este campo é um campo reservado

O valor deste campo está em branco.

O comprimento desse campo é fornecido por MQ\_MCA\_NAME\_LENGTH

#### *MCASecurityId (MQBYTE40)*

Este campo especifica o identificador de segurança para o MCA.

Esse campo não é relevante para os canais com um *ChannelType* de MQCHT\_CLNTCONN

O valor especial a seguir indica que não há identificador de segurança:

#### **MQSID\_NONE**

Nenhum identificador de segurança especificado.

O valor é binário zero para o comprimento do campo.

Para a linguagem de programação C, a constante MQSID\_NONE\_ARRAY também é definida; essa constante possui o mesmo valor que MQSID\_NONE, mas é uma matriz de caracteres em vez de uma sequência.

Este é um campo de entrada / saída na saída. O comprimento desse campo é fornecido por MQ\_SECURITY\_ID\_LENGTH Esse campo não estará presente se o *Version* for menor que MQCD\_VERSION\_6

#### *MCAType (MQLONG)*

Este campo especifica o tipo de programa do agente do canal de mensagens..

Esse campo é relevante apenas para canais com um *ChannelType* de MQCHT\_SENDER, MQCHT\_SERVER, MQCHT\_REQUESTER, MQCHT\_CLUSSDR, ou MQCHT\_CLUSRCVR.

O valor é um dos seguintes:

### **MQMCAT\_PROCESS**

Processo.

O agente do canal de mensagens executado como um processo separado.

### **MQMCAT\_THREAD**

Encadeamento (Multiplataformas).

O agente do canal de mensagens executado como um encadeamento separado.

Esse campo não estará presente quando *Versão* for menor que MQCD\_VERSION\_2.

### *MCAUserIdentifier (MQCHAR12)*

Este campo especifica o identificador de usuários para o MCA (Message Channel Agent).

Este campo usa os primeiros 12 bytes do identificador de usuário MCA e pode ser configurado por um agente de segurança.

Há dois campos que contêm o identificador de usuários do MCA:

- *MCAUserIdentifier* contém os primeiros 12 bytes do identificador de usuário MCA e é preenchido com espaços em branco se o identificador for menor que 12 bytes. *MCAUserIdentifier* pode estar em branco..
- *LongMCAUserIdPtr* aponta para o identificador de usuários do MCA completo, que pode ter mais de 12 bytes.. Seu comprimento é fornecido por *LongMCAUserIdLength*.. O identificador completo não contém espaços em branco finais e não é terminado em nulo. Se o identificador estiver em branco, *LongMCAUserIdLength* será zero e o valor de *LongMCAUserIdPtr* será indefinido.

**Nota:** *LongMCAUserIdPtr* não estará presente se *Version* for menor que MQCD\_VERSION\_6.

Se o identificador de usuário do MCA não estiver em branco, ele especificará o identificador de usuário a ser usado pelo agente do canal de mensagem para autorização para acessar os recursos do IBM MQ . Para os tipos de canal MQCHT\_REQUESTER, MQCHT\_RECEIVER e MQCHT\_CLUSRCVR, se PutAuthority for MQPA\_DEFAULT, este será o identificador de usuário usado para verificações de autorização para a operação put para filas de destino...

Se o identificador de usuário MCA estiver em branco, o agente do canal de mensagens usará seu identificador de usuário padrão.

O identificador de usuários do MCA pode ser configurado por uma saída de segurança para indicar o identificador de usuários que o agente do canal de mensagens deve usar. A saída pode mudar *MCAUserIdentifier* ou a sequência apontada por *LongMCAUserIdPtr*.. Se ambos forem mudados, mas forem diferentes uns dos outros, o MCA usará *LongMCAUserIdPtr* em preferência a *MCAUserIdentifier* Se a saída mudar o comprimento da sequência endereçada por *LongMCAUserIdPtr*, *LongMCAUserIdLength* deverá ser configurado de forma correspondente Se a saída aumentar o comprimento do identificador, a saída deverá alocar armazenamento do comprimento necessário, configurar esse armazenamento para o identificador necessário e colocar o endereço desse armazenamento em *LongMCAUserIdPtr*. A saída é responsável por liberar esse armazenamento quando a saída é chamada posteriormente com a razão MQXR\_TERM

Para canais com um *ChannelType* de MQCHT\_SVRCONN, se *MCAUserIdentifier* na definição de canal estiver em branco, qualquer identificador de usuário transferido do cliente será copiado para ele Esse identificador de usuário (após qualquer modificação pela saída de segurança no servidor) é aquele sob o qual o aplicativo cliente é considerado em execução.

O identificador de usuários MCA não é relevante para canais com um *ChannelType* de MQCHT\_SDR, MQCHT\_SVR, MQCHT\_CLNTCONN, MQCHT\_CLUSSDR.

Este é um campo de entrada / saída na saída. O comprimento desse campo é fornecido por MQ\_USER\_ID\_LENGTH Este campo não estará presente quando *Version* for menor que MQCD\_VERSION\_2.

*ModeName (MQCHAR8)*

Este campo especifica o nome do modo da LU 6.2

Esse campo é relevante apenas se o protocolo da transmissão (*TransportType*) for MQXPT\_LU62e o *ChannelType* não for MQCHT\_SVRCONN ou MQCHT\_RECEIVER.

Este campo está sempre em branco. As informações estão contidas no objeto secundário de comunicações.

O comprimento desse campo é fornecido por MQ\_MODE\_NAME\_LENGTH

*MsgCompLista [16] (MQLONG)*

Este campo especifica a lista de técnicas de compactação de dados da mensagem suportadas pelo canal.

A lista contém um ou mais dos seguintes valores:

**MQCOMPRESS\_NONE**

Nenhuma compactação de dados da mensagem é executada.

**MQCOMPRESS\_RLE**

A compactação de dados da mensagem é executada usando codificação run-length.

**MQCOMPRESS\_ZLIBFAST**

A compactação dos dados da mensagem é feita usando a técnica de compactação zlib. É preferível um tempo de compactação mais rápido.

**MQCOMPRESS\_ZLIBHIGH**

A compactação dos dados da mensagem é feita usando a técnica de compactação zlib. É preferível um nível de compactação alto.

**MQCOMPRESS\_ANY**

Qualquer técnica de compactação suportada pelo gerenciador de filas pode ser usada para a compactação de mensagens MQCOMPRESS\_ANY é válido apenas para canais de receptor, solicitante e conexão do servidor.

**MQCOMPRESS\_NOT\_AVAILABLE**

Os valores não utilizados na lista são configurados para esse valor.

Este é um campo de entrada para a saída O campo não estará presente se *Version* for menor que MQCD\_VERSION\_8.

*MsgExit (MQCHARn)*

Este campo especifica o nome da saída da mensagem do canal.

Se esse nome não estiver em branco, a saída será chamada nos seguintes horários:

- Imediatamente depois que uma mensagem foi obtida da fila de transmissão (remetente ou servidor) ou imediatamente antes de uma mensagem ser colocada em uma fila de destino (receptor ou solicitador).

A saída recebe toda a mensagem de aplicativo e cabeçalho da fila de transmissão para modificação.

- Na inicialização e término do canal.

Esse campo não é relevante para canais com um *ChannelType* de MQCHT\_SVRCONN ou MQCHT\_CLNTCONN; uma saída de mensagens nunca é chamada para esses canais

Consulte “MQCD-Definição de canal” na página 1532 para uma descrição do conteúdo desse campo em vários ambientes.

O comprimento desse campo é fornecido por MQ\_EXIT\_NAME\_LENGTH

**Nota:** O valor dessa constante é específico para o ambiente.

*MsgExitPtr (MQPTR)*

Esse campo especifica o endereço do primeiro campo do *MsgExit*

Se *MsgExitsDefined* for maior que zero, esse endereço será o endereço da lista de nomes de cada saída de mensagem do canal na cadeia.

Cada nome está em um campo de comprimento *ExitNameLength*, preenchido à direita com espaços em branco.. Há *MsgExitsDefined* campos adjacentes um ao outro-um para cada saída.

Quaisquer mudanças feitas nesses nomes por uma saída são preservadas, embora a saída do canal de mensagens não tome nenhuma ação explícita-ela não altera quais saídas são chamadas

Se *MsgExitsDefined* for zero, esse campo será o ponteiro nulo.

Nas plataformas em que a linguagem de programação não suporta o tipo de dados do ponteiro, este campo é declarado como uma cadeia de bytes do comprimento apropriado.

Este é um campo de entrada para a saída O campo não estará presente se *Version* for menor que MQCD\_VERSION\_4..

#### *MsgExitsDefinido (MQLONG)*

Este campo especifica o número de saídas de mensagens de canal definidas na cadeia..

É maior ou igual a zero.

Este é um campo de entrada para a saída O campo não estará presente se *Version* for menor que MQCD\_VERSION\_4..

#### *Contagem de MsgRetry(MQLONG)*

Este campo especifica o número de vezes que o MCA tenta colocar a mensagem, após a primeira tentativa falhar.

Esse campo indica o número de vezes que o MCA tenta a operação open ou put, se o primeiro MQOPEN ou MQPUT falhar com o código de conclusão MQCC\_FAILED. O efeito desse atributo depende se *MsgRetryExit* está em branco ou não em branco:

- Se *MsgRetryExit* estiver em branco, o atributo **MsgRetryCount** controlará se o MCA tenta novamente. Se o valor de atributo for zero, nenhuma tentativa será feita. Se o valor de atributo for maior que zero, as novas tentativas serão feitas em intervalos fornecidos pelo atributo **MsgRetryInterval** .

Novas tentativas são feitas apenas para os seguintes códigos de razão:

- MQRC\_PAGESET\_FULL
- MQRC\_PUT\_INHIBITED
- MQRC\_Q\_FULL

Para outros códigos de razão, o MCA continua imediatamente para seu processamento de falha normal, sem tentar novamente a mensagem com falha

- Se *MsgRetryExit* não estiver em branco, o atributo **MsgRetryCount** não afetará o MCA; em vez disso, é a saída de nova tentativa de mensagem que determina quantas vezes a nova tentativa é tentada e em quais intervalos; a saída é chamada mesmo se o atributo **MsgRetryCount** for zero.

O atributo **MsgRetryCount** é disponibilizado para a saída na estrutura MQCD, mas a saída não é necessária para honrá-la-novas tentativas continuam indefinidamente até que a saída retorne MQXCC\_SUPPRESS\_FUNCTION no campo *ExitResponse* de MQCXP.

Este campo é relevante apenas para canais com um *ChannelType* de MQCHT\_REQUESTER, MQCHT\_RECEIVER, ou MQCHT\_CLUSRCVR

Esse campo não estará presente se *Version* for menor que MQCD\_VERSION\_3

#### *Saída de MsgRetry(MQCHARn)*

Esse campo especifica o nome da saída de nova tentativa de mensagem do canal...

A saída de nova tentativa de mensagem é uma saída chamada pelo MCA quando o MCA recebe um código de conclusão de MQCC\_FAILED de uma chamada MQOPEN ou MQPUT. O propósito da saída é especificar um intervalo de tempo pelo qual o MCA aguarda antes de tentar a operação MQOPEN ou MQPUT novamente. Como alternativa, a saída pode ser configurada para não tentar a operação novamente.

A saída é chamada para todos os códigos de razão que possuem um código de conclusão MQCC\_FAILED- as configurações da saída determinam quais códigos de razão ela deseja que o MCA tente novamente, para quantas tentativas e em quais intervalos de tempo..

Quando a operação não deve ser tentada mais, o MCA executa seu processamento de falha normal; esse processamento inclui gerar uma mensagem de relatório de exceção (se especificado pelo emissor) e colocar a mensagem original na fila de mensagens não entregues ou descartar a mensagem (dependendo se o emissor especificou MQRO\_DEAD\_LETTER\_Q ou MQRO\_DISCARD\_MSG). Falhas envolvendo a fila de mensagens não entregues (por exemplo, fila de mensagens não entregues cheia) não fazem com que a saída de nova tentativa de mensagem seja chamada.

Se o nome da saída não estiver em branco, a saída será chamada nos seguintes horários:

- Imediatamente antes de executar a espera antes de tentar entregar uma mensagem novamente..
- Na inicialização e finalização do canal

Consulte “MQCD-Definição de canal” na página 1532 para uma descrição do conteúdo desse campo em vários ambientes.

Este campo é relevante apenas para canais com um *ChannelType* de MQCHT\_REQUESTER, MQCHT\_RECEIVER, ou MQCHT\_CLUSRCVR

O comprimento desse campo é fornecido por MQ\_EXIT\_NAME\_LENGTH

**Nota:** O valor dessa constante é específico para o ambiente.

Esse campo não estará presente se *Version* for menor que MQCD\_VERSION\_3

#### *Intervalo de MsgRetry(MQLONG)*

Este campo especifica o intervalo mínimo em milissegundos após o qual a operação de abertura ou de colocação é tentada novamente.

O efeito desse atributo depende se *MsgRetryExit* está em branco ou não em branco:

- Se *MsgRetryExit* estiver em branco, o atributo **MsgRetryInterval** especifica o período mínimo que o MCA aguarda antes de tentar novamente uma mensagem, se o primeiro MQOPEN ou MQPUT falhar com o código de conclusão MQCC\_FAILED. Um valor zero significa que a nova tentativa será executada assim que possível após a tentativa anterior. Novas tentativas serão executadas apenas se *MsgRetryCount* for maior que zero.

Esse atributo também é usado como o tempo de espera se a saída de repetição de mensagens retornar um valor inválido no campo *MsgRetryInterval* em MQCXP.

- Se *MsgRetryExit* não estiver em branco, o atributo **MsgRetryInterval** não afetará o MCA; em vez disso, ele é a saída de nova tentativa de mensagem que determina quanto tempo o MCA espera por ele. O atributo **MsgRetryInterval** é disponibilizado para a saída na estrutura MQCD, mas a saída não precisa ser honrada.

O valor está no intervalo de 0 a 999 999 999.

Este campo é relevante apenas para canais com um *ChannelType* de MQCHT\_REQUESTER, MQCHT\_RECEIVER, ou MQCHT\_CLUSRCVR

Esse campo não estará presente se *Version* for menor que MQCD\_VERSION\_3

Os seguintes campos nessa estrutura não estarão presentes se *Version* for menor que MQCD\_VERSION\_4.

#### *MsgRetryUserData (MQCHAR32)*

Este campo especifica os dados do usuário de saída da nova tentativa de mensagem do canal..

Esses dados são passados para a saída de nova tentativa de mensagem do canal no campo *ExitData* do parâmetro **ChannelExitParms** (consulte MQ\_CHANNEL\_EXIT)..

Este campo inicialmente contém os dados que foram configurados na definição de canal... No entanto, durante o tempo de vida dessa instância do MCA, quaisquer mudanças feitas no conteúdo desse campo por uma saída de qualquer tipo são preservadas pelo MCA e visíveis para chamadas subsequentes de



saídas (independentemente do tipo) para essa instância do MCA. Tais mudanças não afetam a definição de canal usada por outras instâncias do MCA.. Qualquer caractere (incluindo dados binários) pode ser usado.

Este campo é relevante apenas para canais com um *ChannelType* de MQCHT\_REQUESTER, MQCHT\_RECEIVER, ou MQCHT\_CLUSRCVR

O comprimento desse campo é fornecido por MQ\_EXIT\_DATA\_LENGTH. Esse campo não estará presente se *Version* for menor que MQCD\_VERSION\_3

Esse campo não é relevante em IBM MQ for IBM i

#### *MsgUser(MQCHAR32)*

Este campo especifica os dados do usuário de saída da mensagem do canal

Esses dados são transmitidos para a saída de mensagem do canal no campo *ExitData* do parâmetro **ChannelExitParms** (consulte MQ\_CHANNEL\_EXIT)..

Este campo inicialmente contém os dados que foram configurados na definição de canal... No entanto, durante o tempo de vida dessa instância do MCA, quaisquer mudanças feitas no conteúdo desse campo por uma saída de qualquer tipo são preservadas pelo MCA e visíveis para chamadas subsequentes de saídas (independentemente do tipo) para essa instância do MCA. Tais mudanças não afetam a definição de canal usada por outras instâncias do MCA.. Qualquer caractere (incluindo dados binários) pode ser usado.

O comprimento desse campo é fornecido por MQ\_EXIT\_DATA\_LENGTH

Esse campo não é relevante em IBM MQ for IBM i

#### *MsgUserDataPtr (MQPTR)*

Esse campo especifica o endereço do primeiro campo do *MsgUserData*

Se *MsgExitsDefined* for maior que zero, esse endereço será o endereço da lista de itens de dados do usuário para cada saída de mensagem de canal na cadeia...

Cada item de dados do usuário está em um campo de comprimento *ExitDataLength*, preenchido à direita com espaços em branco. Há *MsgExitsDefined* campos adjacentes um ao outro-um para cada saída. Se o número de itens de dados do usuário definido for menor do que o número de nomes de saída, os itens de dados do usuário indefinidos serão configurados como espaços em branco. Por outro lado, se o número de itens de dados do usuário definido for maior que o número de nomes de saída, os itens de dados do usuário em excesso serão ignorados e não apresentados à saída.

Todas as mudanças feitas nesses valores por uma saída são preservadas. Isso permite que uma saída passe informações para outra saída. Nenhuma validação é realizada em quaisquer mudanças, portanto, por exemplo, dados binários podem ser gravados nesses campos se necessário.

Se *MsgExitsDefined* for zero, esse campo será o ponteiro nulo.

Nas plataformas em que a linguagem de programação não suporta o tipo de dados do ponteiro, este campo é declarado como uma cadeia de bytes do comprimento apropriado.

Este é um campo de entrada para a saída. O campo não estará presente se *Version* for menor que MQCD\_VERSION\_4..

#### *NetworkPriority (MQLONG)*

Este campo especifica a prioridade da conexão de rede para o canal

Quando vários caminhos para um destino específico estão disponíveis, o caminho com a prioridade mais alta é escolhido. O valor está no intervalo de 0 a 9; 0 é a prioridade mais baixa.

Esse campo é relevante apenas para canais com um *ChannelType* de MQCHT\_CLUSSDR ou MQCHT\_CLUSRCVR..

Este é um campo de entrada para a saída. O campo não estará presente se *Version* for menor que MQCD\_VERSION\_5.

Os campos a seguir nessa estrutura não estarão presentes se *Version* for menor que MQCD\_VERSION\_6.

#### *NonPersistentMsgSpeed (MQLONG)*

Este campo especifica a velocidade na qual mensagens não persistentes viajam pelo canal.

Esse campo é relevante apenas para canais com um *ChannelType* de MQCHT\_SENDER, MQCHT\_SERVER, MQCHT\_RECEIVER, MQCHT\_REQUESTER, MQCHT\_CLUSSDR, ou MQCHT\_CLUSRCVR

O valor é um dos seguintes:

#### **MQNPMS\_NORMAL**

Velocidade normal.

Se um canal for definido como MQNPMS\_NORMAL, as mensagens não persistentes viajam pelo canal em velocidade normal. Isso tem a vantagem de essas mensagens não serem perdidas se houver uma falha de canal. Além disso, as mensagens persistentes e não persistentes na mesma fila de transmissão mantêm sua ordem relativa entre si.

#### **MQNPMS\_FAST**

Velocidade rápida.

Se um canal for definido como MQNPMS\_FAST, as mensagens não persistentes viajam pelo canal em velocidade rápida. Isso melhora o rendimento do canal, mas significa que mensagens não persistentes serão perdidas se houver uma falha do canal. Além disso, é possível que as mensagens não persistentes saltem à frente das mensagens persistentes que aguardam na mesma fila de transmissão, ou seja, a ordem das mensagens não persistentes não é mantida em relação às mensagens persistentes. No entanto, a ordem de mensagens não persistentes relativas umas às outras é mantida. Da mesma forma, a ordem das mensagens persistentes relativas entre si é mantida.

#### *Senha (MQCHAR12)*

Este campo especifica a senha utilizada pelo agente do canal de mensagens ao tentar iniciar uma sessão de SNA segura com um agente do canal de mensagens remoto

Esse campo pode não estar em branco somente no AIX, Linux, and Windowse é relevante apenas para canais com um *ChannelType* de MQCHT\_SENDER, MQCHT\_SERVER, MQCHT\_REQUESTER ou MQCHT\_CLNTCONN.. Em z/OS, esse campo não é relevante

O comprimento desse campo é fornecido por MQ\_PASSWORD\_LENGTH No entanto, apenas os primeiros 10 caracteres são usados..

Esse campo não estará presente se *Version* for menor que MQCD\_VERSION\_2.

#### *PropertyControl (MQLONG)*

Este campo especifica o que acontece às propriedades de mensagens quando a mensagem está prestes a ser enviada para um gerenciador de filas V6 ou anterior (um gerenciador de filas que não entende o conceito de um descritor de propriedade).

O valor pode ser qualquer um dos seguintes valores:

#### **MQPROP\_COMPATIBILITY**

Se a mensagem contiver uma propriedade com um prefixo de **mcd.**, **jms.**, **usr.** ou **mqext.**, todas as propriedades de mensagem serão entregues ao aplicativo em um cabeçalho MQRFH2 . Caso contrário, todas as propriedades da mensagem, exceto aquelas contidas no descritor de mensagens (ou extensão), serão descartadas e não estarão mais acessíveis ao aplicativo.

Esse valor é o valor padrão; ele permite que os aplicativos, que esperam que as propriedades relacionadas ao JMS estejam em um cabeçalho MQRFH2 nos dados da mensagem, continuem a trabalhar sem modificação....

#### **MQPROP\_NONE**

Todas as propriedades da mensagem, exceto aquelas propriedades no descritor de mensagem (ou extensão), são removidas da mensagem antes que a mensagem seja enviada ao gerenciador de filas remotas.

## **MQPROP\_ALL**

Todas as propriedades da mensagem são incluídas com a mensagem quando ela é enviada para o gerenciador de filas remotas. As propriedades, exceto aquelas propriedades no descritor de mensagens (ou extensão), são colocadas em um ou mais cabeçalhos MQRFH2 nos dados da mensagem.

Esse atributo é aplicável aos canais Emissor, Servidor, Emissor de Cluster e Receptor de Cluster.

[“MQIA\\_\\* \(Seletores de atributo de número inteiro\)” na página 130](#)

[“MQPROP\\_\\* \(Valores de Controle de Propriedade de Fila e Canal e Comprimento Máximo de Propriedades\)” na página 170](#)

### *PutAuthority (MQLONG)*

Este campo especifica se o identificador de usuário nas informações de contexto associadas a uma mensagem é utilizado para estabelecer autoridade para colocar a mensagem na fila de destino.

Este campo é relevante apenas para canais com um *ChannelType* de MQCHT\_REQUESTER, MQCHT\_RECEIVER, ou MQCHT\_CLUSRCVR. É um dos seguintes:

### **MQPA\_DEFAULT**

O identificador de usuário padrão é usado.

### **MQPA\_CONTEXT**

O identificador de usuário de contexto é usado.

### **MQPA\_ALTERNATE\_OR\_MCA**

O ID do usuário do campo *UserIdentifier* do descritor da mensagem é usado. Qualquer ID de usuário recebido da rede é utilizado. Este valor é suportado apenas no z/OS.

### **MQPA\_ONLY\_MCA**

O ID padrão de usuário é utilizado. Qualquer ID de usuário recebido da rede é utilizado. Este valor é suportado apenas no z/OS.

### *QMGrName (MQCHAR48)*

Esse campo especifica o nome do gerenciador de filas ao qual uma saída pode se conectar.

Para canais com um *ChannelType* diferente de MQCHT\_CLNTCONN, esse campo é o nome do gerenciador de fila ao qual uma saída pode se conectar, que no AIX, Linux, and Windows, sempre não está em branco.

O comprimento desse campo é fornecido por MQ\_Q\_MGR\_NAME\_LENGTH.

### *ReceiveExit (MQCHARn)*

Este campo especifica o nome da saída de recebimento do canal.

Se esse nome não estiver em branco, a saída será chamada nos seguintes horários:

- Imediatamente antes dos dados de rede recebidos serem processados.

A saída recebe o buffer de transmissão completo conforme recebido. O conteúdo do buffer pode ser modificado conforme requerido.

- Na inicialização e término do canal.

Consulte [“MQCD-Definição de canal” na página 1532](#) para uma descrição do conteúdo desse campo em vários ambientes.

O comprimento desse campo é fornecido por MQ\_EXIT\_NAME\_LENGTH.

**Nota:** O valor dessa constante é específico para o ambiente.

### *ReceiveExitPtr (MQPTR)*

Esse campo especifica o endereço do primeiro campo do *ReceiveExit*.

Se *ReceiveExitsDefined* for maior que zero, esse endereço será o endereço da lista de nomes de cada saída de recebimento de canal na cadeia.

Cada nome está em um campo de comprimento *ExitNameLength*, preenchido à direita com espaços em branco.. Há *ReceiveExitsDefined* campos adjacentes um ao outro-um para cada saída.

Quaisquer mudanças feitas nesses nomes por uma saída são preservadas, embora a saída do canal de mensagens não tome nenhuma ação explícita-ela não altera quais saídas são chamadas

Se *ReceiveExitsDefined* for zero, esse campo será o ponteiro nulo.

Nas plataformas em que a linguagem de programação não suporta o tipo de dados do ponteiro, este campo é declarado como uma cadeia de bytes do comprimento apropriado.

Este é um campo de entrada para a saída O campo não estará presente se *Version* for menor que MQCD\_VERSION\_4..

#### *ReceiveExitsDefinido (MQLONG)*

Este campo especifica o número de saídas de recebimento do canal definidas na cadeia..

É maior ou igual a zero.

Este é um campo de entrada para a saída O campo não estará presente se *Version* for menor que MQCD\_VERSION\_4..

#### *Dados de ReceiveUser(MQCHAR32)*

Este canal especifica os dados do usuário de saída de recebimento do canal

Esses dados são passados para a saída de recebimento do canal no campo *ExitData* do parâmetro **ChannelExitParms** (consulte MQ\_CHANNEL\_EXIT)..

Este campo inicialmente contém os dados que foram configurados na definição de canal... No entanto, durante o tempo de vida dessa instância do MCA, quaisquer mudanças feitas no conteúdo desse campo por uma saída de qualquer tipo são preservadas pelo MCA e visíveis para chamadas subsequentes de saídas (independentemente do tipo) para essa instância do MCA. Isso se aplica a saídas em conversas diferentes.. Tais mudanças não afetam a definição de canal usada por outras instâncias do MCA.. Qualquer caractere (incluindo dados binários) pode ser usado.

O comprimento desse campo é fornecido por MQ\_EXIT\_DATA\_LENGTH

Esse campo não é relevante em IBM MQ for IBM i

Os campos a seguir nessa estrutura não estarão presentes se *Version* for menor que MQCD\_VERSION\_2.

#### *ReceiveUserDataPtr (MQPTR)*

Esse campo especifica o endereço do primeiro campo do *ReceiveUserData*

Se *ReceiveExitsDefined* for maior que zero, esse endereço será o endereço da lista de itens de dados do usuário para cada saída de recebimento de canal na cadeia..

Cada item de dados do usuário está em um campo de comprimento *ExitDataLength*, preenchido à direita com espaços em branco Há *ReceiveExitsDefined* campos adjacentes um ao outro-um para cada saída. Se o número de itens de dados do usuário definido for menor do que o número de nomes de saída, os itens de dados do usuário indefinidos serão configurados como espaços em branco Por outro lado, se o número de itens de dados do usuário definido for maior que o número de nomes de saída, os itens de dados do usuário em excesso serão ignorados e não apresentados à saída.

Todas as mudanças feitas nesses valores por uma saída são preservadas Isso permite que uma saída passe informações para outra saída. Nenhuma validação é realizada em quaisquer mudanças, portanto, por exemplo, dados binários podem ser gravados nesses campos se necessário.

Se *ReceiveExitsDefined* for zero, esse campo será o ponteiro nulo.

Nas plataformas em que a linguagem de programação não suporta o tipo de dados do ponteiro, este campo é declarado como uma cadeia de bytes do comprimento apropriado.

Este é um campo de entrada para a saída O campo não estará presente se *Version* for menor que MQCD\_VERSION\_4..

Os seguintes campos nessa estrutura não estarão presentes se *Version* for menor que MQCD\_VERSION\_5.

#### *RemotePassword* (MQCHAR12)

Este campo especifica a senha de um parceiro.

Esse campo conterá informações válidas apenas se *ChannelType* for MQCHT\_CLNTCONN ou MQCHT\_SVRCONN

- Para uma saída de segurança em um canal MQCHT\_CLNTCONN, essa senha é uma senha obtida do ambiente. A saída pode optar por enviá-la para a saída de segurança no servidor
- Para uma saída de segurança em um canal MQCHT\_SVRCONN, esse campo pode conter uma senha que foi obtida do ambiente no cliente, se não houver saída de segurança do cliente. A saída pode usar essa senha para validar o identificador de usuário em *RemoteUserIdentifier*..

Se houver uma saída de segurança no cliente, essas informações poderão ser obtidas em um fluxo de segurança do cliente.

O comprimento desse campo é fornecido por MQ\_PASSWORD\_LENGTH Esse campo não estará presente se *Version* for menor que MQCD\_VERSION\_2.

#### *RemoteSecurityId* (MQBYTE40)

Este campo especifica o identificador de segurança do usuário remoto.

Esse campo é relevante apenas para canais com um *ChannelType* de MQCHT\_CLNTCONN ou MQCHT\_SVRCONN

O valor especial a seguir indica que não há identificador de segurança:

#### **MQSID\_NONE**

Nenhum identificador de segurança especificado.

O valor é binário zero para o comprimento do campo.

Para a linguagem de programação C, a constante MQSID\_NONE\_ARRAY também é definida; essa constante possui o mesmo valor que MQSID\_NONE, mas é uma matriz de caracteres em vez de uma sequência.

Este é um campo de entrada para a saída O comprimento desse campo é fornecido por MQ\_SECURITY\_ID\_LENGTH Esse campo não estará presente se o *Version* for menor que MQCD\_VERSION\_6

Os seguintes campos nessa estrutura não estarão presentes se *Version* for menor que MQCD\_VERSION\_7.

#### *RemoteUserIdentificador* (MQCHAR12)

Este campo especifica os primeiros 12 bytes de um identificador de usuário de um parceiro.

Há dois campos que contêm o identificador de usuário remoto:

- *RemoteUserIdentifier* contém os primeiros 12 bytes do identificador de usuário remoto e é preenchido com espaços em branco se o identificador for menor que 12 bytes. *RemoteUserIdentifier* pode estar em branco..
- *LongRemoteUserIdPtr* aponta para o identificador de usuário remoto completo, que pode ter mais de 12 bytes. Seu comprimento é fornecido por *LongRemoteUserIdLength*.. O identificador completo não contém espaços em branco finais e não é terminado em nulo. Se o identificador estiver em branco, *LongRemoteUserIdLength* será zero e o valor de *LongRemoteUserIdPtr* será indefinido.

*LongRemoteUserIdPtr* não estará presente se *Version* for menor que MQCD\_VERSION\_6.

O identificador de usuário remoto é relevante apenas para canais com um *ChannelType* de MQCHT\_CLNTCONN ou MQCHT\_SVRCONN.

- Para uma saída de segurança em um canal MQCHT\_CLNTCONN, esse valor é um identificador de usuário que foi obtido do ambiente... A saída pode optar por enviá-la para a saída de segurança no servidor
- Para uma saída de segurança em um canal MQCHT\_SVRCONN, esse campo pode conter um identificador de usuário que foi obtido do ambiente no cliente, se não houver saída de segurança do cliente. A saída pode validar esse ID de usuário (possivelmente com a senha em *RemotePassword*) e atualizar o valor em *MCAUserIdentifier*.

Se houver uma saída de segurança no cliente, essas informações poderão ser obtidas em um fluxo de segurança do cliente.

O comprimento desse campo é fornecido por MQ\_USER\_ID\_LENGTH. Esse campo não estará presente se *Version* for menor que MQCD\_VERSION\_2.

#### *SecurityExit (MQCHARn)*

Este campo especifica o nome da saída de segurança do canal

Se esse nome não estiver em branco, a saída será chamada nos seguintes horários:

- Imediatamente após estabelecer um canal.

Antes de quaisquer mensagens serem transferidas, é concedida à saída a oportunidade de instigar fluxos de segurança para validar a autorização de conexão.

- No recebimento de uma resposta para um fluxo de mensagem de segurança.

Quaisquer fluxos de mensagens de segurança recebidos do processador remoto na máquina remota são fornecidos à saída.

- Na inicialização e término do canal.

Consulte [“MQCD-Definição de canal”](#) na página 1532 para uma descrição do conteúdo desse campo em vários ambientes.

O comprimento desse campo é fornecido por MQ\_EXIT\_NAME\_LENGTH

**Nota:** O valor dessa constante é específico para o ambiente.

#### *Dados de SecurityUser(MQCHAR32)*

Este canal especifica os dados do usuário de saída de segurança do canal

Esses dados são transmitidos à saída de segurança do canal no campo *ExitData* do parâmetro **ChannelExitParms** (consulte MQ\_CHANNEL\_EXIT).

Este campo inicialmente contém os dados que foram configurados na definição de canal... No entanto, durante o tempo de vida dessa instância do MCA, quaisquer mudanças feitas no conteúdo desse campo por uma saída de qualquer tipo são preservadas pelo MCA e visíveis para chamadas subsequentes de saídas (independentemente do tipo) para essa instância do MCA. Isso se aplica a saídas em conversas diferentes.. Tais mudanças não afetam a definição de canal usada por outras instâncias de MCA. Qualquer caractere (incluindo dados binários) pode ser usado.

O comprimento desse campo é fornecido por MQ\_EXIT\_DATA\_LENGTH

Esse campo não é relevante em IBM MQ for IBM i

#### *SendExit (MQCHARn)*

Esse campo especifica o nome da saída de envio do canal..

Se esse nome não estiver em branco, a saída será chamada nos seguintes horários:

- Imediatamente antes dos dados serem enviados na rede.

É concedido à saída o buffer de transmissão completo antes de ser transmitido. O conteúdo do buffer pode ser modificado conforme requerido.

- Na inicialização e término do canal.

Consulte “MQCD-Definição de canal” na página 1532 para uma descrição do conteúdo desse campo em vários ambientes.

O comprimento desse campo é fornecido por MQ\_EXIT\_NAME\_LENGTH

**Nota:** O valor dessa constante é específico para o ambiente.

*SendExitPtr (MQPTR)*

Esse campo especifica o endereço do primeiro campo do *SendExit*

Se *SendExitsDefined* for maior que zero, esse endereço será o endereço da lista de nomes de cada canal de saída de envio na cadeia...

Cada nome está em um campo de comprimento *ExitNameLength*, preenchido à direita com espaços em branco.. Há *SendExitsDefined* campos adjacentes um ao outro-um para cada saída.

Quaisquer mudanças feitas nesses nomes por uma saída são preservadas, embora a saída de envio de mensagem não tome nenhuma ação explícita-ela não altera quais saídas são chamadas

Se *SendExitsDefined* for zero, esse campo será o ponteiro nulo.

Nas plataformas em que a linguagem de programação não suporta o tipo de dados do ponteiro, este campo é declarado como uma cadeia de bytes do comprimento apropriado.

Este é um campo de entrada para a saída O campo não estará presente se *Version* for menor que MQCD\_VERSION\_4..

*SendExitsDefinido (MQLONG)*

Este campo especifica o número de saídas de envio do canal definidas na cadeia..

É maior ou igual a zero.

Este é um campo de entrada para a saída O campo não estará presente se *Version* for menor que MQCD\_VERSION\_4..

*Dados de SendUser(MQCHAR32).*

Este campo especifica os dados do usuário de saída de envio do canal

Esses dados são transmitidos para a saída de envio do canal no campo *ExitData* do parâmetro **ChannelExitParms** (consulte MQ\_CHANNEL\_EXIT)..

Este campo inicialmente contém os dados que foram configurados na definição de canal... No entanto, durante o tempo de vida dessa instância do MCA, quaisquer mudanças feitas no conteúdo desse campo por uma saída de qualquer tipo são preservadas pelo MCA e visíveis para chamadas subsequentes de saídas (independentemente do tipo) para essa instância do MCA. Isso se aplica a saídas em conversas diferentes.. Tais mudanças não afetam a definição de canal usada por outras instâncias do MCA.. Qualquer caractere (incluindo dados binários) pode ser usado.

O comprimento desse campo é fornecido por MQ\_EXIT\_DATA\_LENGTH

Esse campo não é relevante em IBM MQ for IBM i

*SendUserDataPtr (MQPTR)*

Esse campo especifica o endereço do campo *SendUserData* ..

Se *SendExitsDefined* for maior que zero, esse endereço será o endereço da lista de itens de dados do usuário para cada saída de mensagem de canal na cadeia...

Cada item de dados do usuário está em um campo de comprimento *ExitDataLength*, preenchido à direita com espaços em branco Há *MsgExitsDefined* campos adjacentes um ao outro-um para cada saída. Se o número de itens de dados do usuário definido for menor do que o número de nomes de saída, os itens de dados do usuário indefinidos serão configurados como espaços em branco Por outro lado, se o número de itens de dados do usuário definido for maior que o número de nomes de saída, os itens de dados do usuário em excesso serão ignorados e não apresentados à saída.

Todas as mudanças feitas nesses valores por uma saída são preservadas. Isso permite que uma saída passe informações para outra saída. Nenhuma validação é realizada em quaisquer mudanças, portanto, por exemplo, dados binários podem ser gravados nesses campos se necessário.

Se *SendExitsDefined* for zero, esse campo será o ponteiro nulo.

Nas plataformas em que a linguagem de programação não suporta o tipo de dados do ponteiro, este campo é declarado como uma cadeia de bytes do comprimento apropriado.

Este é um campo de entrada para a saída. O campo não estará presente se *Version* for menor que MQCD\_VERSION\_4..

#### *SeqNumberWrap (MQLONG)*

Este campo especifica o número de sequência de mensagens mais alto permitido.

Quando este valor é atingido, os números de sequência retornam para o início novamente em 1.

Este valor não é negociável e deve corresponder às definições dos canais local e remoto.

Esse campo não é relevante para canais com um *ChannelType* de MQCHT\_SVRCON ou MQCHT\_CLNTCONN.

#### *Compartilhamento de Conversas (MQLONG)*

Este campo especifica o número máximo de conversas que podem compartilhar uma instância do canal associada a este canal..

Esse campo é usado na conexão do cliente e nos canais de conexão do servidor.

Um valor de 0 significa que o canal opera como fez em versões anteriores a IBM WebSphere MQ 7.0 com relação aos atributos a seguir:

- Compartilhamento de conversa
- Ler antes
- STOP CHANNEL (*channelname*) MODE (QUIESCE)
- Pulsação
- Consumo Assíncrono do Cliente

Um valor 1 é o valor mínimo para o comportamento IBM MQ. Embora apenas uma conversa seja permitida na instância do canal, a leitura antecipada, o consumo assíncrono e o comportamento de CLNTCONN-SVRCONN pulsação e parada de canal quiescente estão disponíveis.

Este é um campo de entrada para a saída. Não estará presente se *Version* for menor que MQCD\_VERSION\_9.

O valor padrão desse campo é 10.

**Nota:** *MaxInstances* e *MaxInstancesPerClient* limites aplicados a um canal restringem o número de instâncias do canal, não o número de conversas que podem estar compartilhando essas instâncias.

#### *ShortConnectionNome (MQCHAR20)*

Este campo especifica os primeiros 20 bytes de um nome de conexão.

Se o campo *Version* for MQCD\_VERSION\_1, *ShortConnectionName* conterá o nome completo da conexão.

Se o campo *Version* for MQCD\_VERSION\_2 ou superior, *ShortConnectionName* conterá os primeiros 20 caracteres do nome da conexão. O nome completo da conexão é fornecido pelo campo *ConnectionName*; *ShortConnectionName* e os primeiros 20 caracteres de *ConnectionName* são idênticos..

Consulte *ConnectionName* para obter detalhes sobre o conteúdo deste campo.

**Nota:** O nome desse campo foi mudado para MQCD\_VERSION\_2 e versões subsequentes do MQCD; o campo foi chamado anteriormente *ConnectionName*.



O comprimento desse campo é fornecido por MQ\_SHORT\_CONN\_NAME\_LENGTH

#### *ShortRetryContagem (MQLONG)*

Este campo especifica o número máximo de tentativas feitas para conectar a uma máquina remota.

Esse campo é o número máximo de tentativas feitas para se conectar à máquina remota, em intervalos especificados por *ShortRetryInterval*, antes que *LongRetryCount* e *LongRetryInterval* (normalmente mais longos) sejam usados.

Esse campo é relevante apenas para canais com um *ChannelType* de MQCHT\_SENDER, MQCHT\_SERVER, MQCHT\_CLUSSDR, ou MQCHT\_CLUSRCVR

#### *ShortRetryIntervalo (MQLONG)*

Este campo especifica o número máximo de segundos a aguardar, antes de tentar novamente a conexão com a máquina remota

O intervalo entre as repetições poderá ser estendido, se o canal tiver que esperar para tornar-se ativo.

Esse campo é relevante apenas para canais com um *ChannelType* de MQCHT\_SENDER, MQCHT\_SERVER, MQCHT\_CLUSSDR, ou MQCHT\_CLUSRCVR

#### *SPLProtection (MQLONG)*

Este campo especifica o valor da proteção de política de segurança AMS .

O valor é um dos seguintes:

#### **MQSPL\_PASSTHRU**

Passar, inalterado, quaisquer mensagens enviadas ou recebidas pelo MCA para este canal.

Esse valor é relevante apenas para canais com um *ChannelType* de MQCHT\_SENDER, MQCHT\_SERVER, MQCHT\_RECEIVER, ou MQCHT\_REQUESTER e é o valor padrão.

#### **MQSPL\_REMOVE**

Remova qualquer proteção do AMS de mensagens recuperadas da fila de transmissão pelo MCA e envie as mensagens para o parceiro.

Esse valor é relevante apenas para canais com um *ChannelType* de MQCHT\_SENDER ou MQCHT\_SERVER.

#### **MQSPL\_ ASPOLICY**

Com base na política definida para a fila de destino, aplique a proteção do AMS nas mensagens de entrada antes de colocá-las na fila de destino.

Esse valor é relevante apenas para canais com um *ChannelType* de MQCHT\_RECEIVER ou MQCHT\_REQUESTER.

Este é um campo de entrada para a saída. Esse campo não estará presente se o *Version* for menor que MQCD\_VERSION\_12

#### *SSLCipherSpec (MQCHAR32)*

Este campo especifica a Especificação de Cipher que está em uso ao usar TLS

Se SSLCipherSpec estiver em branco, o canal não está usando TLS. Se não estiver em branco, esse campo conterá uma sequência especificando o CipherSpec em uso.

Esse parâmetro é válido para todos os tipos de canal. Ele é suportado nas seguintes plataformas:

-  AIX
-  IBM i
-  Linux
-  Windows
-  z/OS

Ele é válido apenas para tipos de canal de um tipo de transporte (TRPTYPE) de TCP

Este é um campo de entrada para a saída O comprimento desse campo é fornecido por MQ\_SSL\_CIPHER\_SPEC\_LENGTH O campo não estará presente se *Version* for menor que MQCD\_VERSION\_7.

*SSLClientAuth (MQLONG)*

Este campo especifica se a autenticação de cliente TLS é necessária

Este campo é relevante apenas para definições de canal SVRCONN.

Ele é um dos seguintes valores:

**MQSCA\_REQUIRED**

Autenticação de cliente necessária.

**MQSCA\_OPTIONAL**

Autenticação de cliente opcional.

Este é um campo de entrada para a saída O campo não estará presente se *Version* for menor que MQCD\_VERSION\_7.

*SSLPeerNameComprimento (MQLONG)*

Esse campo especifica o comprimento em bytes do nome do peer TLS apontado por *SSLPeerNamePtr*.

Este é um campo de entrada para a saída O campo não estará presente se *Version* for menor que MQCD\_VERSION\_7.

*SSLPeerNamePtr (MQPTR)*

Este campo especifica o endereço do nome do peer do TLS

Quando um certificado é recebido durante um handshake TLS bem-sucedido, o Nome Distinto do assunto do certificado é copiado no campo MQCD acessado por *SSLPeerNamePtr* no final do canal que recebe o certificado. Ele sobrescreve o valor *SSLPeerName* para o canal se esse valor estiver presente na definição de canal do usuário local. Se uma saída de segurança for especificada nessa extremidade do canal, ela receberá o Nome Distinto do certificado peer no MQCD.

Este é um campo de entrada para a saída O campo não estará presente se *Version* for menor que MQCD\_VERSION\_7.

**Nota:** Aplicativos de saída de segurança construídos antes da liberação do IBM WebSphere MQ 7.1 podem requerer atualização. Para obter mais informações, consulte [Programas de saída de segurança de canal](#).

*StrucLength (MQLONG)*

Este campo especifica o comprimento em bytes da estrutura MQCD.

O comprimento não inclui nenhuma das cadeias endereçadas por campos de ponteiro contidos dentro da estrutura. O valor é um dos seguintes:

**MQCD\_LENGTH\_4**

Comprimento da estrutura de definição de canal version-4 .

**MQCD\_LENGTH\_5**

Comprimento da estrutura de definição de canal version-5 .

**MQCD\_LENGTH\_6**

Comprimento da estrutura de definição de canal version-6 .

**MQCD\_LENGTH\_7**

Comprimento da estrutura de definição de canal version-7 .

**MQCD\_LENGTH\_8**

Comprimento da estrutura de definição de canal version-8 .

**MQCD\_LENGTH\_9**

Comprimento da estrutura de definição de canal version-9 .

**MQCD\_LENGTH\_10**

Comprimento da estrutura de definição de canal version-10 .

**MQCD\_LENGTH\_11**

Comprimento da estrutura de definição de canal version-11 .

**z/OS MQCD\_LENGTH\_12**

Comprimento da estrutura de definição de canal version-12 .

A constante a seguir especifica o comprimento da versão atual:

**MQCD\_CURRENT\_LENGTH**

Comprimento da versão atual da estrutura de definição de canal.

**Nota:** Essas constantes possuem valores específicos do ambiente.

O campo não estará presente se *Version* for menor que MQCD\_VERSION\_4..

*TpName (MQCHAR64)*

Esse campo especifica o nome do programa de transação da LU 6.2 .

Esse campo é relevante apenas se o protocolo da transmissão (*TransportType*) for MQXPT\_LU62e o *ChannelType* não for MQCHT\_SVRCONN ou MQCHT\_RECEIVER.

Esse campo está sempre em branco em plataformas nas quais as informações estão contidas no Objeto Lado de comunicações...

O comprimento desse campo é fornecido por MQ\_TP\_NAME\_LENGTH

*TransportType (MQLONG)*

Este campo especifica o protocolo de transmissão a ser utilizado

O valor não será verificado se o canal foi iniciado da outra extremidade.

Ele é um dos seguintes valores:

**MQXPT\_LU62**

protocolo de transporte LU 6.2 .

**MQXPT\_TCP**

protocolo de transporte TCP/IP.

**MQXPT\_NETBIOS**

Protocolo de transporte do NetBIOS

Esse valor é suportado nos seguintes ambientes: Windows.

**MQXPT\_SPX**

Protocolo de transporte SPX.

Esse valor é suportado nos seguintes ambientes: Windows, mais IBM MQ clientes conectados a esses sistemas.

*UseDLQ (MQLONG)*

Este campo especifica se a fila de mensagens não entregues (ou fila de mensagens não entregues) é usada quando as mensagens não podem ser entregues por canais.

Ele pode conter um dos seguintes valores:

**MQUSEDLQ\_NO**

Mensagens que não podem ser entregues por um canal são tratadas como uma falha. O canal descarta a mensagem ou o canal termina, de acordo com a configuração NPMSPEED.

**MQUSEDLQ\_YES**

Quando o atributo do gerenciador de filas DEADQ fornece o nome de uma fila de mensagens não entregues, ele é usado, caso contrário, o comportamento é como para NO YES é o valor padrão.

### *UserIdentifier* (MQCHAR12)

Este campo especifica o identificador de usuário utilizado pelo agente do canal de mensagem ao tentar iniciar uma sessão SNA segura com um agente do canal de mensagens remoto..

Esse campo pode não estar em branco somente no AIX, Linux, and Windowse é relevante apenas para canais com um *ChannelType* de MQCHT\_SENDER, MQCHT\_SERVER, MQCHT\_REQUESTER ou MQCHT\_CLNTCONN.. Em z/OS, esse campo não é relevante

O comprimento desse campo é fornecido por MQ\_USER\_ID\_LENGTH No entanto, apenas os primeiros 10 caracteres são usados..

Este campo não estará presente quando *Version* for menor que MQCD\_VERSION\_2.

### *Versão* (MQLONG)

O campo *Version* especifica o número da versão mais alto que você pode configurar para a estrutura

O valor depende do ambiente:

#### **MQCD\_VERSION\_1**

Estrutura de definição de canal da Versão 1

#### **MQCD\_VERSION\_2**

Estrutura de definição de canal da versão 2.

#### **MQCD\_VERSION\_3**

Estrutura de definição de canal da Versão 3

#### **MQCD\_VERSION\_4**

Estrutura de definição de canal da Versão 4

#### **MQCD\_VERSION\_5**

Estrutura de definição de canal da Versão 5

#### **MQCD\_VERSION\_6**

Estrutura de definição de canal da Versão 6

#### **MQCD\_VERSION\_7**

Estrutura de definição de canal da Versão 7

#### **MQCD\_VERSION\_8**

Estrutura de definição de canal da Versão 8

#### **MQCD\_VERSION\_9**

Estrutura de definição de canal da Versão 9

#### **MQCD\_VERSION\_10**

Estrutura de definição de canal da Versão 10

#### **MQCD\_VERSION\_11**

Estrutura de definição de canal da Versão 11

A versão 11 é a mais alta para a qual é possível configurar o campo no IBM MQ 8.0 em todas as plataformas

#### **MQCD\_VERSION\_12**

Estrutura de definição de canal da Versão 12

A versão 12 é a mais alta para a qual é possível configurar o campo em IBM MQ 9.1.3

Os campos que existem apenas nas versões mais recente da estrutura são identificados como tal nas descrições dos campos. A constante a seguir especifica o número da versão atual:

#### **MQCD\_CURRENT\_VERSION**

O valor configurado em MQCD\_CURRENT\_VERSION é a versão atual da estrutura de definição de canal sendo usada.

O valor de MQCD\_CURRENT\_VERSION depende do ambiente. Ele contém o valor mais alto suportado pela plataforma.

MQCD\_CURRENT\_VERSION não é usado para inicializar as estruturas padrão fornecidas nos arquivos de cabeçalho, cópia e inclusão fornecidos para diferentes linguagens de programação. A inicialização padrão do Version depende da plataforma e da liberação..

As declarações MQCD no cabeçalho, cópia e arquivos include são inicializados para MQCD\_VERSION\_6. Para usar campos MQCD adicionais, os aplicativos devem configurar o número da versão como MQCD\_CURRENT\_VERSION. Se você estiver gravando um aplicativo que seja móvel entre vários ambientes, deverá escolher uma versão que seja suportada em todos os ambientes.

**Sugestão:** Quando uma nova versão da estrutura MQCD é introduzida, o layout da parte existente não é alterado.. A saída deve verificar o número da versão Deve ser igual ou maior que a versão mais baixa que contém os campos que a saída precisa usar.

#### *XmitQName (MQCHAR48)*

Esse campo especifica o nome da fila de transmissão da qual as mensagens são recuperadas.

Esse campo é relevante apenas para canais que possuem um *ChannelType* de MQCHT\_SENDER ou MQCHT\_SERVER

O comprimento desse campo é fornecido por MQ\_Q\_NAME\_LENGTH

### **Declaração C**

Esta é a declaração C para a estrutura MQCD.

```
typedef struct tagMQCD MQCD;
typedef MQCD MQPOINTER PMQCD;
typedef PMQCD MQPOINTER PPMQCD;

struct tagMQCD {
    MQCHAR    ChannelName[20];           /* Channel definition name */
    MQLONG    Version;                  /* Structure version number */
    MQLONG    ChannelType;              /* Channel type */
    MQLONG    TransportType;            /* Transport type */
    MQCHAR    Desc[64];                 /* Channel description */
    MQCHAR    QMgrName[48];             /* Queue manager name */
    MQCHAR    XmitQName[48];           /* Transmission queue name */
    MQCHAR    ShortConnectionName[20]; /* First 20 bytes of */
                                           /* connection name */
    MQCHAR    MCAName[20];              /* Reserved */
    MQCHAR    ModeName[8];              /* LU 6.2 Mode name */
    MQCHAR    TpName[64];               /* LU 6.2 transaction program */
                                           /* name */
    MQLONG    BatchSize;                 /* Batch size */
    MQLONG    DiscInterval;             /* Disconnect interval */
    MQLONG    ShortRetryCount;          /* Short retry count */
    MQLONG    ShortRetryInterval;       /* Short retry wait interval */
    MQLONG    LongRetryCount;           /* Long retry count */
    MQLONG    LongRetryInterval;        /* Long retry wait interval */
    MQCHAR    SecurityExit[128];        /* Channel security exit name */
    MQCHAR    MsgExit[128];             /* Channel message exit name */
    MQCHAR    SendExit[128];           /* Channel send exit name */
    MQCHAR    ReceiveExit[128];        /* Channel receive exit name */
    MQLONG    SeqNumberWrap;            /* Highest allowable message */
                                           /* sequence number */
    MQLONG    MaxMsgLength;             /* Maximum message length */
    MQLONG    PutAuthority;             /* Put authority */
    MQLONG    DataConversion;           /* Data conversion */
    MQCHAR    SecurityUserData[32];     /* Channel security exit user */
                                           /* data */
    MQCHAR    MsgUserData[32];          /* Channel message exit user */
                                           /* data */
    MQCHAR    SendUserData[32];        /* Channel send exit user */
                                           /* data */
    MQCHAR    ReceiveUserData[32];     /* Channel receive exit user */
                                           /* data */
    /* Ver:1 */
    MQCHAR    UserIdentifier[12];       /* User identifier */
    MQCHAR    Password[12];            /* Password */
    MQCHAR    MCAUserIdentifier[12];   /* First 12 bytes of MCA user */
                                           /* identifier */
    MQLONG    MCAType;                  /* Message channel agent type */
    MQCHAR    ConnectionName[264];     /* Connection name */
    MQCHAR    RemoteUserIdentifier[12]; /* First 12 bytes of user */
                                           /* identifier from partner */
}
```

```

MQCHAR RemotePassword[12]; /* Password from partner */
/* Ver:2 */
MQCHAR MsgRetryExit[128]; /* Channel message retry exit */
/* name */
MQCHAR MsgRetryUserData[32]; /* Channel message retry exit */
/* user data */
MQLONG MsgRetryCount; /* Number of times MCA will */
/* try to put the message, */
/* after first attempt has */
/* failed */
MQLONG MsgRetryInterval; /* Minimum interval in */
/* milliseconds after which */
/* the open or put operation */
/* will be retried */

/* Ver:3 */
MQLONG HeartbeatInterval; /* Time in seconds between */
/* heartbeat flows */
MQLONG BatchInterval; /* Batch duration */
MQLONG NonPersistentMsgSpeed; /* Speed at which */
/* nonpersistent messages are */
/* sent */
MQLONG StrucLength; /* Length of MQCD structure */
MQLONG ExitNameLength; /* Length of exit name */
MQLONG ExitDataLength; /* Length of exit user data */
MQLONG MsgExitsDefined; /* Number of message exits */
/* defined */
MQLONG SendExitsDefined; /* Number of send exits */
/* defined */
MQLONG ReceiveExitsDefined; /* Number of receive exits */
/* defined */
MQPTR MsgExitPtr; /* Address of first MsgExit */
/* field */
MQPTR MsgUserDataPtr; /* Address of first */
/* MsgUserData field */
MQPTR SendExitPtr; /* Address of first SendExit */
/* field */
MQPTR SendUserDataPtr; /* Address of first */
/* SendUserData field */
MQPTR ReceiveExitPtr; /* Address of first */
/* ReceiveExit field */
MQPTR ReceiveUserDataPtr; /* Address of first */
/* ReceiveUserData field */

/* Ver:4 */
MQPTR ClusterPtr; /* Address of a list of */
/* cluster names */
MQLONG ClustersDefined; /* Number of clusters to */
/* which the channel belongs */
MQLONG NetworkPriority; /* Network priority */

/* Ver:5 */
MQLONG LongMCAUserIdLength; /* Length of long MCA user */
/* identifier */
MQLONG LongRemoteUserIdLength; /* Length of long remote user */
/* identifier */
MQPTR LongMCAUserIdPtr; /* Address of long MCA user */
/* identifier */
MQPTR LongRemoteUserIdPtr; /* Address of long remote */
/* user identifier */
MQBYTE40 MCASecurityId; /* MCA security identifier */
MQBYTE40 RemoteSecurityId; /* Remote security identifier */

/* Ver:6 */
MQCHAR SSLCipherSpec[32]; /* TLS CipherSpec */
MQPTR SSLPeerNamePtr; /* Address of TLS peer name */
MQLONG SSLPeerNameLength; /* Length of TLS peer name */
MQLONG SSLClientAuth; /* Whether TLS client */
/* authentication is required */
MQLONG KeepAliveInterval; /* Keepalive interval */
MQCHAR LocalAddress[48]; /* Local communications */
/* address */
MQLONG BatchHeartbeat; /* Batch heartbeat interval */

/* Ver:7 */
MQLONG HdrCompList[2]; /* Header data compression */
/* list */
MQLONG MsgCompList[16]; /* Message data compression */
/* list */
MQLONG CLWLChannelRank; /* Channel rank */
MQLONG CLWLChannelPriority; /* Channel priority */
MQLONG CLWLChannelWeight; /* Channel weight */
MQLONG ChannelMonitoring; /* Channel monitoring */
MQLONG ChannelStatistics; /* Channel statistics */

/* Ver:8 */
MQLONG SharingConversations; /* Limit on sharing */
/* conversations */

```

```

MQLONG   PropertyControl;           /* Message property control */
MQLONG   MaxInstances;             /* Limit on SVRCONN channel */
MQLONG   MaxInstancesPerClient;    /* instances */
MQLONG   MaxInstancesPerClient;    /* Limit on SVRCONN channel */
MQLONG   ClientChannelWeight;      /* instances per client */
MQLONG   ConnectionAffinity;       /* Client channel weight */
/* Ver:9 */
MQLONG   BatchDataLimit;           /* Connection affinity */
MQLONG   UseDLQ;                   /* Batch data limit */
MQLONG   DefReconnect;             /* Use Dead Letter Queue */
/* option */
/* Ver:10 */
MQCHAR64 CertificateLabel;         /* Default client reconnect */
/* Ver:11 */
MQLONG   SPLProtection             /* Certificate label */
/* Ver:12 */
};

```

## Declaração COBOL

Esta declaração é a declaração de COBOL para a estrutura MQCD

```

** MQCD structure
  10 MQCD.
    ** Channel definition name
    15 MQCD-CHANNELNAME PIC X(20).
    ** Structure version number
    15 MQCD-VERSION PIC S9(9) BINARY.
    ** Channel type
    15 MQCD-CHANNELTYPE PIC S9(9) BINARY.
    ** Transport type
    15 MQCD-TRANSPORTTYPE PIC S9(9) BINARY.
    ** Channel description
    15 MQCD-DESC PIC X(64).
    ** Queue manager name
    15 MQCD-QMGRNAME PIC X(48).
    ** Transmission queue name
    15 MQCD-XMITQNAME PIC X(48).
    ** First 20 bytes of connection name
    15 MQCD-SHORTCONNECTIONNAME PIC X(20).
    ** Reserved
    15 MQCD-MCANAME PIC X(20).
    ** LU 6.2 Mode name
    15 MQCD-MODENAME PIC X(8).
    ** LU 6.2 transaction program name
    15 MQCD-TPNAME PIC X(64).
    ** Batch size
    15 MQCD-BATCHSIZE PIC S9(9) BINARY.
    ** Disconnect interval
    15 MQCD-DISCINTERVAL PIC S9(9) BINARY.
    ** Short retry count
    15 MQCD-SHORTRETRYCOUNT PIC S9(9) BINARY.
    ** Short retry wait interval
    15 MQCD-SHORTRETRYINTERVAL PIC S9(9) BINARY.
    ** Long retry count
    15 MQCD-LONGRETRYCOUNT PIC S9(9) BINARY.
    ** Long retry wait interval
    15 MQCD-LONGRETRYINTERVAL PIC S9(9) BINARY.
    ** Channel security exit name
    15 MQCD-SECURITYEXIT PIC X(20).
    ** Channel message exit name
    15 MQCD-MSGEXIT PIC X(20).
    ** Channel send exit name
    15 MQCD-SENDEXIT PIC X(20).
    ** Channel receive exit name
    15 MQCD-RECEIVEEXIT PIC X(20).
    ** Highest allowable message sequence number
    15 MQCD-SEQNUMBERWRAP PIC S9(9) BINARY.
    ** Maximum message length
    15 MQCD-MAXMSGLENGTH PIC S9(9) BINARY.
    ** Put authority
    15 MQCD-PUTAUTHORITY PIC S9(9) BINARY.
    ** Data conversion
    15 MQCD-DATACONVERSION PIC S9(9) BINARY.
    ** Channel security exit user data
    15 MQCD-SECURITYUSERDATA PIC X(32).
    ** Channel message exit user data
    15 MQCD-MSGUSERDATA PIC X(32).

```

```

** Channel send exit user data
  15 MQCD-SENDUSERDATA PIC X(32).
** Channel receive exit user data
  15 MQCD-RECEIVEUSERDATA PIC X(32).
** Ver:1 **
** User identifier
  15 MQCD-USERIDENTIFIER PIC X(12).
** Password
  15 MQCD-PASSWORD PIC X(12).
** First 12 bytes of MCA user identifier
  15 MQCD-MCAUSERIDENTIFIER PIC X(12).
** Message channel agent type
  15 MQCD-MCATYPE PIC S9(9) BINARY.
** Connection name
  15 MQCD-CONNECTIONNAME PIC X(264).
** First 12 bytes of user identifier from partner
  15 MQCD-REMOTEUSERIDENTIFIER PIC X(12).
** Password from partner
  15 MQCD-REMOTEPASSWORD PIC X(12).
** Ver:2 **
** Channel message retry exit name
  15 MQCD-MSGRETRYEXIT PIC X(20).
** Channel message retry exit user data
  15 MQCD-MSGRETRYUSERDATA PIC X(32).
** Number of times MCA will try to put the message, after first
** attempt has failed
  15 MQCD-MSGRETRYCOUNT PIC S9(9) BINARY.
** Minimum interval in milliseconds after which the open or put
** operation will be retried
  15 MQCD-MSGRETRYINTERVAL PIC S9(9) BINARY.
** Ver:3 **
** Time in seconds between heartbeat flows
  15 MQCD-HEARTBEATINTERVAL PIC S9(9) BINARY.
** Batch duration
  15 MQCD-BATCHINTERVAL PIC S9(9) BINARY.
** Speed at which nonpersistent messages are sent
  15 MQCD-NONPERSISTENTMSGSPPEED PIC S9(9) BINARY.
** Length of MQCD structure
  15 MQCD-STRUCLLENGTH PIC S9(9) BINARY.
** Length of exit name
  15 MQCD-EXITNAMELENGTH PIC S9(9) BINARY.
** Length of exit user data
  15 MQCD-EXITDATALENGTH PIC S9(9) BINARY.
** Number of message exits defined
  15 MQCD-MSGEXITSDEFINED PIC S9(9) BINARY.
** Number of send exits defined
  15 MQCD-SENDEXITSDEFINED PIC S9(9) BINARY.
** Number of receive exits defined
  15 MQCD-RECEIVEEXITSDEFINED PIC S9(9) BINARY.
** Address of first MsgExit field
  15 MQCD-MSGEXITPTR POINTER.
** Address of first MsgUserData field
  15 MQCD-MSGUSERDATAPTR POINTER.
** Address of first SendExit field
  15 MQCD-SENDEXITPTR POINTER.
** Address of first SendUserData field
  15 MQCD-SENDUSERDATAPTR POINTER.
** Address of first ReceiveExit field
  15 MQCD-RECEIVEEXITPTR POINTER.
** Address of first ReceiveUserData field
  15 MQCD-RECEIVEUSERDATAPTR POINTER.
** Ver:4 **
** Address of a list of cluster names
  15 MQCD-CLUSTERPTR POINTER.
** Number of clusters to which the channel belongs
  15 MQCD-CLUSTERSDEFINED PIC S9(9) BINARY.
** Network priority
  15 MQCD-NETWORKPRIORITY PIC S9(9) BINARY.
** Ver:5 **
** Length of long MCA user identifier
  15 MQCD-LONGMCAUSERIDLENGTH PIC S9(9) BINARY.
** Length of long remote user identifier
  15 MQCD-LONGREMOTEUSERIDLENGTH PIC S9(9) BINARY.
** Address of long MCA user identifier
  15 MQCD-LONGMCAUSERIDPTR POINTER.
** Address of long remote user identifier
  15 MQCD-LONGREMOTEUSERIDPTR POINTER.
** MCA security identifier
  15 MQCD-MCASECURITYID PIC X(40).
** Remote security identifier
  15 MQCD-REMOTECURITYID PIC X(40).
** Ver:6 **

```



```

** TLS CipherSpec
  15 MQCD-SSLCIPHERSPEC PIC X(32).
** Address of TLS peer name
  15 MQCD-SSLPEERNAMEPTR POINTER.
** Length of TLS peer name
  15 MQCD-SSLPEERNAMELENGTH PIC S9(9) BINARY.
** Whether TLS client authentication is required
  15 MQCD-SSLCLIENTAUTH PIC S9(9) BINARY.
** Keepalive interval
  15 MQCD-KEEPALIVEINTERVAL PIC S9(9) BINARY.
** Local communications address
  15 MQCD-LOCALADDRESS PIC X(48).
** Batch heartbeat interval
  15 MQCD-BATCHHEARTBEAT PIC S9(9) BINARY.
** Ver:7 **
** Header data compression list
  15 MQCD-HDRCOMPLIST PIC S9(9) BINARY.
** Message data compression list
  15 MQCD-MSGCOMPLIST PIC S9(9) BINARY.
** Channel rank
  15 MQCD-CLWLCHANNELRANK PIC S9(9) BINARY.
** Channel priority
  15 MQCD-CLWLCHANNELPRIORITY PIC S9(9) BINARY.
** Channel weight
  15 MQCD-CLWLCHANNELWEIGHT PIC S9(9) BINARY.
** Channel monitoring
  15 MQCD-CHANNELMONITORING PIC S9(9) BINARY.
** Channel statistics
  15 MQCD-CHANNELSTATISTICS PIC S9(9) BINARY.
** Ver:8 **
** Limit on sharing conversations
  15 MQCD-SHARINGCONVERSATIONS PIC S9(9) BINARY.
** Message property control
  15 MQCD-PROPERTYCONTROL PIC S9(9) BINARY.
** Limit on SVRCONN channel instances
  15 MQCD-MAXINSTANCES PIC S9(9) BINARY.
** Limit on SVRCONN channel instances per client
  15 MQCD-MAXINSTANCESPERCLIENT PIC S9(9) BINARY.
** Client channel weight
  15 MQCD-CLIENTCHANNELWEIGHT PIC S9(9) BINARY.
** Connection affinity
  15 MQCD-CONNECTIONAFFINITY PIC S9(9) BINARY.
** Ver:9 **
** Batch data limit
  15 MQCD-BATCHDATALIMIT PIC S9(9) BINARY.
** Use Dead Letter Queue
  15 MQCD-USEDLQ PIC S9(9) BINARY.
** Default client reconnect option
  15 MQCD-DEFRECONNECT PIC S9(9) BINARY.
** Ver:10 **
** Certificate Label
  15 MQCD-CERTLABL PIC X (64)
** Ver:11 **
** AMS Security policy protection
  15 MQCD-SPLPROTECTION PIC S9(9) BINARY
** Ver:12 **

```

## **Declaração RPG (ILE)**

Esta é a declaração RPG para a estrutura MQCD.

```

D* MQCD Structure
D*
D* Channel definition name
D CDCHN          1      20
D* Structure version number
D CDVER          21     24I 0
D* Channel type
D CDCHT          25     28I 0
D* Transport type
D CDTRT          29     32I 0
D* Channel description
D CDDDES         33     96
D* Queue manager name
D CDQM           97     144
D* Transmission queue name
D CDXQ          145     192
D* First 20 bytes of connection name
D CDSCN         193     212

```

```

D* Reserved
D CDMCA 213 232
D* LU 6.2 Mode name
D CDMOD 233 240
D* LU 6.2 transaction program name
D CDTP 241 304
D* Batch size
D CDBS 305 308I 0
D* Disconnect interval
D CDDI 309 312I 0
D* Short retry count
D CDSRC 313 316I 0
D* Short retry wait interval
D CDSRI 317 320I 0
D* Long retry count
D CDLRC 321 324I 0
D* Long retry wait interval
D CDLRI 325 328I 0
D* Channel security exit name
D CDSCX 329 348
D* Channel message exit name
D CDMSX 349 368
D* Channel send exit name
D CDSNX 369 388
D* Channel receive exit name
D CDRCX 389 408
D* Highest allowable message sequence number
D CDSNW 409 412I 0
D* Maximum message length
D CDMML 413 416I 0
D* Put authority
D CDPA 417 420I 0
D* Data conversion
D CDDC 421 424I 0
D* Channel security exit user data
D CDSCD 425 456
D* Channel message exit user data
D CDMSD 457 488
D* Channel send exit user data
D CDSND 489 520
D* Channel receive exit user data
D CDRCU 521 552
D* Ver:1 **
D* User identifier
D CDUID 553 564
D* Password
D CDPW 565 576
D* First 12 bytes of MCA user identifier
D CDAUI 577 588
D* Message channel agent type
D CDCAT 589 592I 0
D* Connection name
D CDCON 593 848
D CDCN2 849 856
D* First 12 bytes of user identifier from partner
D CDRUI 857 868
D* Password from partner
D CDRPW 869 880
D* Ver:2 **
D* Channel message retry exit name
D CDMRX 881 900
D* Channel message retry exit user data
D CDMRD 901 932
D* Number of times MCA will try to put the message, after first
D* attempt has failed
D CDMRC 933 936I 0
D* Minimum interval in milliseconds after which the open or put
D* operation will be retried
D CDMRI 937 940I 0
D* Ver:3 **
D* Time in seconds between heartbeat flows
D CDHBI 941 944I 0
D* Batch duration
D CDBI 945 948I 0
D* Speed at which nonpersistent messages are sent
D CDNPM 949 952I 0
D* Length of MQCD structure
D CDLEN 953 956I 0
D* Length of exit name
D CDXNL 957 960I 0
D* Length of exit user data
D CDXDL 961 964I 0

```

```

D* Number of message exits defined
D CDMXD          965    968I 0
D* Number of send exits defined
D CDSXD          969    972I 0
D* Number of receive exits defined
D CDRXD          973    976I 0
D* Address of first MsgExit field
D CDMXP          977    992*
D* Address of first MsgUserData field
D CDMUP          993    1008*
D* Address of first SendExit field
D CDSXP          1009   1024*
D* Address of first SendUserData field
D CDSUP          1025   1040*
D* Address of first ReceiveExit field
D CDRXP          1041   1056*
D* Address of first ReceiveUserData field
D CDRUP          1057   1072*
D* Ver:4 **
D* Address of a list of cluster names
D CDCLP          1073   1088*
D* Number of clusters to which the channel belongs
D CDCLD          1089   1092I 0
D* Network priority
D CDNP          1093   1096I 0
D* Ver:5 **
D* Length of long MCA user identifier
D CDML          1097   1100I 0
D* Length of long remote user identifier
D CDRL          1101   1104I 0
D* Address of long MCA user identifier
D CDLMP         1105   1120*
D* Address of long remote user identifier
D CDLRP         1121   1136*
D* MCA security identifier
D CDMSI         1137   1176
D* Remote security identifier
D CDRSI         1177   1216
D* Ver:6 **
D* TLS CipherSpec
D CDSCS         1217   1248
D* Address of TLS peer name
D CDSPN         1249   1264*
D* Length of TLS peer name
D CDSPL         1265   1268I 0
D* Whether TLS client authentication is required
D CDSCA         1269   1272I 0
D* Keepalive interval
D CDKAI         1273   1276I 0
D* Local communications address
D CDLOA         1277   1324
D* Batch heartbeat interval
D CDBHB         1325   1328I 0
D* Ver:7 **
D* Header data compression list
D CDHCL0
D CDHCL1         1329   1332I 0
D CDHCL2         1333   1336I 0
D CDHCL          10I 0 DIM(2) OVERLAY(CDHCL0)
D* Message data compression list
D CDMCL0
D CDMCL1         1337   1340I 0
D CDMCL2         1341   1344I 0
D CDMCL3         1345   1348I 0
D CDMCL4         1349   1352I 0
D CDMCL5         1353   1356I 0
D CDMCL6         1357   1360I 0
D CDMCL7         1361   1364I 0
D CDMCL8         1365   1368I 0
D CDMCL9         1369   1372I 0
D CDMCL10        1373   1376I 0
D CDMCL11        1377   1380I 0
D CDMCL12        1381   1384I 0
D CDMCL13        1385   1388I 0
D CDMCL14        1389   1392I 0
D CDMCL15        1393   1396I 0
D CDMCL16        1397   1400I 0
D CDMCL          10I 0 DIM(16) OVERLAY(CDMCL0)
D* Channel rank
D CDCWCR        1401   1404I 0
D* Channel priority
D CDCWCP        1405   1408I 0

```

```

D* Channel weight
D CDCWCW          1409  1412I 0
D* Channel monitoring
D CDCHLMON        1413  1416I 0
D* Channel statistics
D CDCHLST         1417  1420I 0
D* Ver:8 **
D* Limit on sharing conversations
D CDSHC           1421  1424I 0
D* Message property control
D CDPRC           1425  1428I 0
D* Limit on SVRCONN channel instances
D CDMXIN          1429  1432I 0
D* Limit on SVRCONN channel instances per client
D CDMXIC          1433  1436I 0
D* Client channel weight
D CDCLNCHLW      1437  1440I 0
D* Connection affinity
D CDCONNAFF      1441  1444I 0
D* Ver:9 **
D* Batch data limit
D CDBDL          1445  1448I 0
D* Use Dead Letter Queue
D CDUDLQ         1449  1452I 0
D* Default client reconnect option
D CDDRCN         1453  1456I 0
D* Ver:10 **

```

## Declaração do assembler System/390

Essa declaração é a declaração do assembler System/390 para a estrutura MQCD..

```

MQCD              DSECT
MQCD_CHANNELNAME DS CL20  Channel definition name
MQCD_VERSION      DS F      Structure version number
MQCD_CHANNELTYPE  DS F      Channel type
MQCD_TRANSPORTTYPE DS F      Transport type
MQCD_DESC         DS CL64  Channel description
MQCD_QMGRNAME     DS CL48  Queue manager name
MQCD_XMITQNAME    DS CL48  Transmission queue name
MQCD_SHORTCONNECTIONNAME DS CL20 First 20 bytes of connection
* name
MQCD_MCANAME      DS CL20  Reserved
MQCD_MODENAME     DS CL8   LU 6.2 Mode name
MQCD_TPNAME       DS CL64  LU 6.2 transaction program name
MQCD_BATCHSIZE    DS F      Batch size
MQCD_DISCINTERVAL DS F      Disconnect interval
MQCD_SHORTRETRYCOUNT DS F      Short retry count
MQCD_SHORTRETRYINTERVAL DS F      Short retry wait interval
MQCD_LONGRETRYCOUNT DS F      Long retry count
MQCD_LONGRETRYINTERVAL DS F      Long retry wait interval
MQCD_SECURITYEXIT DS CLn   Channel security exit name
MQCD_MSGEXIT      DS CLn   Channel message exit name
MQCD_SENDEXIT     DS CLn   Channel send exit name
MQCD_RECEIVEEXIT  DS CLn   Channel receive exit name
MQCD_SEQNUMBERWRAP DS F      Highest allowable message
* sequence number
MQCD_MAXMSGLLENGTH DS F      Maximum message length
MQCD_PUTAUTHORITY DS F      Put authority
MQCD_DATACONVERSION DS F      Data conversion
MQCD_SECURITYUSERDATA DS CL32 Channel security exit user data
MQCD_MSGUSERDATA  DS CL32 Channel message exit user data
MQCD_SENDUSERDATA DS CL32 Channel send exit user data
MQCD_RECEIVEUSERDATA DS CL32 Channel receive exit user data
MQCD_USERIDENTIFIER DS CL12 User identifier
MQCD_PASSWORD     DS CL12 Password
MQCD_MCAUSERIDENTIFIER DS CL12 First 12 bytes of MCA user
* identifier
MQCD_MCATYPE      DS F      Message channel agent type
MQCD_CONNECTIONNAME DS CL264 Connection name
MQCD_REMOTEUSERIDENTIFIER DS CL12 First 12 bytes of user
* identifier from partner
MQCD_REMOTEPASSWORD DS CL12 Password from partner
MQCD_MSGRETRYEXIT DS CLn   Channel message retry exit name
MQCD_MSGRETRYUSERDATA DS CL32 Channel message retry exit user
* data
MQCD_MSGRETRYCOUNT DS F      Number of times MCA will try to
* put the message, after the
* first attempt has failed

```

MQCD_MSGRETRYINTERVAL	DS	F	Minimum interval in milliseconds after which the open or put operation will be retried
*			
MQCD_HEARTBEATINTERVAL	DS	F	Time in seconds between heartbeat flows
*			
MQCD_BATCHINTERVAL	DS	F	Batch duration
MQCD_NONPERSISTENTMSGSPEED	DS	F	Speed at which nonpersistent messages are sent
*			
MQCD_STRUCLNGTH	DS	F	Length of MQCD structure
MQCD_EXITNAMELENGTH	DS	F	Length of exit name
MQCD_EXITDATALENGTH	DS	F	Length of exit user data
MQCD_MSGEXITSDEFINED	DS	F	Number of message exits defined
MQCD_SENDEXITSDEFINED	DS	F	Number of send exits defined
MQCD_RECEIVEEXITSDEFINED	DS	F	Number of receive exits defined
MQCD_MSGEXITPTR	DS	F	Address of first MSGEXIT field
MQCD_MSGUSERDATAPTR	DS	F	Address of first MSGUSERDATA field
*			
MQCD_SENDEXITPTR	DS	F	Address of first SENDEXIT field
MQCD_SENDUSERDATAPTR	DS	F	Address of first SENDUSERDATA field
*			
MQCD_RECEIVEEXITPTR	DS	F	Address of first RECEIVEEXIT field
*			
MQCD_RECEIVEUSERDATAPTR	DS	F	Address of first RECEIVEUSERDATA field
*			
MQCD_CLUSTERPTR	DS	F	Address of a list of cluster names
*			
MQCD_CLUSTERSDEFINED	DS	F	Number of clusters to which the channel belongs
*			
MQCD_NETWORKPRIORITY	DS	F	Network priority
MQCD_LONGMCAUSERIDLENGTH	DS	F	Length of long MCA user identifier
*			
MQCD_LONGREMOTEUSERIDLENGTH	DS	F	Length of long remote user identifier
*			
MQCD_LONGMCAUSERIDPTR	DS	F	Address of long MCA user identifier
*			
MQCD_LONGREMOTEUSERIDPTR	DS	F	Address of long remote user identifier
*			
MQCD_MCASECURITYID	DS	XL40	MCA security identifier
MQCD_REMOTECURITYID	DS	XL40	Remote security identifier
MQCD_SSLCIPHERSPEC	DS	CL32	TLS CipherSpec
MQCD_SSLPEERNAMEPTR	DS	F	Address of TLS peer name
MQCD_SSLPEERNAMELENGTH	DS	F	Length of TLS peer name
MQCD_SSLCLIENTAUTH	DS	F	Whether TLS client authentication is required
*			
MQCD_KEEPLIVEINTERVAL	DS	F	Keepalive interval
MQCD_LOCALADDRESS	DS	CL48	Local communications address
MQCD_BATCHHEARTBEAT	DS	F	Batch heartbeat interval
MQCD_HDRCOMPLIST	DS	CL2	Header data compression list
MQCD_MSGCOMPLIST	DS	CL16	Message data compression list
MQCD_CLWLCHANNELRANK	DS	F	Channel rank
MQCD_CLWLCHANNELPRIORITY	DS	F	Channel priority
MQCD_CLWLCHANNELWEIGHT	DS	F	Channel weight
MQCD_CHANNELMONITORING	DS	F	Channel monitoring
MQCD_CHANNELSTATISTICS	DS	F	Channel statistics
MQCD_SHARINGCONVERSATIONS	DS	F	Limit on sharing conversations
*			
MQCD_PROPERTYCONTROL	DS	F	Message property control
*			
MQCD_SHARINGCONVERSATIONS	DS	F	Limit on sharing conversations
MQCD_PROPERTYCONTROL	DS	F	Message property control
MQCD_MAXINSTANCES	DS	F	Limit on SVRCONN chl instances
MQCD_MAXINSTANCESPERCLIENT	DS	F	Limit on SVRCONN chl instances per client
MQCD_CLIENTCHANNELWEIGHT	DS	F	Channel weight
MQCD_CONNECTIONAFFINITY	DS	F	Connection Affinty
MQCD_BATCHDATALIMIT	DS	F	Batch data limit
MQCD_USEDLQ	DS	F	Use dead-letter queue
MQCD_DEFRECONNECT	DS	F	Default client reconnect option
MQCD_CERTLABL	DS	F	Certificate label
MQCD_SPLPROTECTION	DS	F	AMS Security policy protection
MQCD_LENGTH	EQU	*-MQCD	
	ORG	MQCD	
MQCD_AREA	DS	CL (MQCD_LENGTH)	

### **Declaração do Visual Basic**

Esta declaração é a declaração Visual Basic da estrutura MQCD.

No Visual Basic, a estrutura MQCD pode ser usada com a estrutura MQCNO na chamada MQCONN.

Type MQCD		
ChannelName	As String*20	'Channel definition name'
Version	As Long	'Structure version number'
ChannelType	As Long	'Channel type'
TransportType	As Long	'Transport type'
Desc	As String*64	'Channel description'
QMgrName	As String*48	'Queue manager name'
XmitQName	As String*48	'Transmission queue name'
ShortConnectionName	As String*20	'First 20 bytes of connection' 'name'
MCAName	As String*20	'Reserved'
ModeName	As String*8	'LU 6.2 Mode name'
TpName	As String*64	'LU 6.2 transaction program name'
BatchSize	As Long	'Batch size'
DiscInterval	As Long	'Disconnect interval'
ShortRetryCount	As Long	'Short retry count'
ShortRetryInterval	As Long	'Short retry wait interval'
LongRetryCount	As Long	'Long retry count'
LongRetryInterval	As Long	'Long retry wait interval'
SecurityExit	As String*128	'Channel security exit name'
MsgExit	As String*128	'Channel message exit name'
SendExit	As String*128	'Channel send exit name'
ReceiveExit	As String*128	'Channel receive exit name'
SeqNumberWrap	As Long	'Highest allowable message' 'sequence number'
MaxMsgLength	As Long	'Maximum message length'
PutAuthority	As Long	'Put authority'
DataConversion	As Long	'Data conversion'
SecurityUserData	As String*32	'Channel security exit user data'
MsgUserData	As String*32	'Channel message exit user data'
SendUserData	As String*32	'Channel send exit user data'
ReceiveUserData	As String*32	'Channel receive exit user data'
UserIdentifier	As String*12	'User identifier'
Password	As String*12	'Password'
MCAUserIdentifier	As String*12	'First 12 bytes of MCA user' 'identifier'
MCAType	As Long	'Message channel agent type'
ConnectionName	As String*264	'Connection name'
RemoteUserIdentifier	As String*12	'First 12 bytes of user' 'identifier from partner'
RemotePassword	As String*12	'Password from partner'
MsgRetryExit	As String*128	'Channel message retry exit name'
MsgRetryUserData	As String*32	'Channel message retry exit user' 'data'
MsgRetryCount	As Long	'Number of times MCA will try to' 'put the message, after the' 'first attempt has failed'
MsgRetryInterval	As Long	'Minimum interval in' 'milliseconds after which the' 'open or put operation will be' 'retried'
HeartbeatInterval	As Long	'Time in seconds between' 'heartbeat flows'
BatchInterval	As Long	'Batch duration'
NonPersistentMsgSpeed	As Long	'Speed at which nonpersistent' 'messages are sent'
StrucLength	As Long	'Length of MQCD structure'
ExitNameLength	As Long	'Length of exit name'
ExitDataLength	As Long	'Length of exit user data'
MsgExitsDefined	As Long	'Number of message exits defined'
SendExitsDefined	As Long	'Number of send exits defined'
ReceiveExitsDefined	As Long	'Number of receive exits defined'
MsgExitPtr	As MQPTR	'Address of first MsgExit field'
MsgUserDataPtr	As MQPTR	'Address of first MsgUserData' 'field'
SendExitPtr	As MQPTR	'Address of first SendExit field'
SendUserDataPtr	As MQPTR	'Address of first SendUserData' 'field'
ReceiveExitPtr	As MQPTR	'Address of first ReceiveExit' 'field'
ReceiveUserDataPtr	As MQPTR	'Address of first' 'ReceiveUserData field'
ClusterPtr	As MQPTR	'Address of a list of cluster' 'names'
ClustersDefined	As Long	'Number of clusters to which the' 'channel belongs'
NetworkPriority	As Long	'Network priority'
LongMCAUserIdLength	As Long	'Length of long MCA user' 'identifier'
LongRemoteUserIdLength	As Long	'Length of long remote user'

LongMCAUserIdPtr	As MQPTR	'identifier' 'Address of long MCA user'
LongRemoteUserIdPtr	As MQPTR	'identifier' 'Address of long remote user'
MCASecurityId	As MQBYTE40	'MCA security identifier'
RemoteSecurityId	As MQBYTE40	'Remote security identifier'
SSLCipherSpec	As String*32	'TLS CipherSpec'
SSLPeerNamePtr	As MQPTR	'Address of TLS peer name'
SSLPeerNameLength	As Long	'Length of TLS peer name'
SSLClientAuth	As Long	'Whether TLS client' 'authentication is required'
KeepAliveInterval	As Long	'Keepalive interval'
LocalAddress	As String*48	'Local communications address'
BatchHeartbeat	As Long	'Batch heartbeat interval'
HdrCompList(0 to 1)	As Long2	'Header data compression list'
MsgCompList(0 To 15)	As Long16	'Message data compression list'
CLWLChannelRank	As Long	'Channel Rank'
CLWLChannelPriority	As Long	'Channel priority'
CLWLChannelWeight	As Long	'Channel Weight'
ChannelMonitoring	As Long	'Channel Monitoring control'
ChannelStatistics	As Long	'Channel Statistics'
End Type		

### **Mudando campos MQCD em uma saída de canal**

Uma saída do canal pode mudar os campos no MQCD. No entanto, essas mudanças geralmente não são realizadas, exceto nas circunstâncias listadas.

Se um programa de saída do canal mudar um campo na estrutura de dados MQCD, o novo valor geralmente será ignorado pelo processo do canal IBM MQ. No entanto, o novo valor permanece no MQCD e é passado a qualquer saída restante em uma sequência de saída e a qualquer conversa que compartilhando instância do canal.

Se SharingConversations for configurado como FALSE na estrutura MQCXP, as mudanças em determinados campos poderão ser atuadas, dependendo do tipo de programa de saída, do tipo de canal e do código de razão de saída. A tabela a seguir mostra os campos que podem ser alterados e afetar o comportamento do canal e em quais circunstâncias. Se um programa de saída alterar um desses campos em qualquer outra circunstância ou qualquer campo não listado, o novo valor será ignorado pelo processo do canal. O novo valor permanece no MQCD e é transmitido para quaisquer saídas restantes em uma cadeia de saída e para qualquer conversa que compartilhar a instância do canal.

Qualquer tipo de programa de saída quando chamado para inicialização (MQXR\_INIT) pode alterar o campo ChannelName de qualquer tipo de canal, desde que MQCXP SharingConversations seja configurado como FALSE. Somente uma saída de segurança pode alterar o campo MCAUserIdentifier, independentemente do valor de MQCXP SharingConversations.

<b>Campo</b>	<b>Código de razão de saída</b>	<b>Tipo de saída.</b>	<b>Tipo de canal</b>
ChannelName	MQXR_INIT	Todos(as)	Todos(as)
TransportType	MQXR_INIT	Todos(as)	Todos(as)
XmitQName	MQXR_INIT	Todos(as)	SDR, RCVR
ModeName	MQXR_INIT	Todos(as)	Todos(as)
TpName	MQXR_INIT	Todos(as)	Todos(as)
BatchSize	MQXR_INIT	Todos(as)	SDR, SVR, RCVR, RQSTR, CLUSSDR, CLUSRCVR

Tabela 823. Campos que podem ser mudados e afetar o comportamento do canal (continuação)

<b>Campo</b>	<b>Código de razão de saída</b>	<b>Tipo de saída.</b>	<b>Tipo de canal</b>
DiscInterval	MQXR_INIT	Todos(as)	SDR, SVR, RCVR, RQSTR, CLUSSDR, CLUSRCVR
ShortRetryContagem	MQXR_INIT	Todos(as)	SDR, SVR, RCVR, RQSTR, CLUSSDR, CLUSRCVR
shortRetryInterval	MQXR_INIT	Todos(as)	SDR, SVR, RCVR, RQSTR, CLUSSDR, CLUSRCVR
LongRetryContagem	MQXR_INIT	Todos(as)	SDR, SVR, RCVR, RQSTR, CLUSSDR, CLUSRCVR
longRetryInterval	MQXR_INIT	Todos(as)	SDR, SVR, RCVR, RQSTR, CLUSSDR, CLUSRCVR
Agrupamento de SeqNumber	MQXR_INIT	Todos(as)	SDR, SVR, RCVR, RQSTR, CLUSSDR, CLUSRCVR
MaxMsgLength	MQXR_INIT	Todos(as)	Todos(as)
PutAuthority	MQXR_INIT	Todos(as)	SDR, SVR, RCVR, RQSTR, CLUSSDR, CLUSRCVR
DataConversion	MQXR_INIT	Todos(as)	Todos(as)
MCAUserIdentifier	MQXR_INIT, MQXR_INIT_SEC, MQXR_SEC_MSG, MQXR_SEC_PARMS	Segurança	RCVR, RQSTR, SVRCONN, CLUSRCVR
ConnectionName	MQXR_INIT	Todos(as)	SDR, SVR, RQSTR, CLNTCONN e CLUSSDR e CLUSRCVR



Tabela 823. Campos que podem ser mudados e afetar o comportamento do canal (continuação)

<b>Campo</b>	<b>Código de razão de saída</b>	<b>Tipo de saída.</b>	<b>Tipo de canal</b>
MsgRetryUserData	MQXR_INIT	Todos(as)	RCVR, RQSTR, CLUSRCVR
Contagem de MsgRetry	MQXR_INIT	Todos(as)	RCVR, RQSTR, CLUSRCVR
Intervalo de MsgRetry	MQXR_INIT	Todos(as)	RCVR, RQSTR, CLUSRCVR
HeartbeatInterval	MQXR_INIT	Todos(as)	Todos(as)
BatchInterval	MQXR_INIT	Todos(as)	SDR, SVR, CLUSSDR, CLUSRCVR
NonPersistentMsgSpeed	MQXR_INIT	Todos(as)	SDR, SVR, RCVR, RQSTR, CLUSSDR, CLUSRCVR
MCASecurityId	MQXR_INIT, MQXR_INIT_SEC, MQXR_SEC_MSG, MQXR_SEC_PARMS	Segurança	SDR, SVR, RCVR, RQSTR e SVRCONN, CLUSSDR e CLUSRCVR
SSLCipherSpec	MQXR_INIT	Todos(as)	Todos(as)
SSLPeerNamePtr	MQXR_INIT	Todos(as)	Todos(as)
SSLPeerNameComprimen to	MQXR_INIT	Todos(as)	Todos(as)
SSLClientAuth	MQXR_INIT	Todos(as)	SVR, RCVR, RQSTR, SVRCONN, CLUSRCVR
KeepAliveInterval	MQXR_INIT	Todos(as)	Todos(as)
LocalAddress	MQXR_INIT	Todos(as)	SDR, SVR, RQSTR, CLNTCONN e CLUSSDR e CLUSRCVR
BatchHeartbeat	MQXR_INIT	Todos(as)	SDR, SVR, CLUSSDR, CLUSRCVR
HdrCompList	MQXR_INIT	Todos(as)	Todos(as)
MsgCompList	MQXR_INIT	Todos(as)	Todos(as)

Tabela 823. Campos que podem ser mudados e afetar o comportamento do canal (continuação)

<b>Campo</b>	<b>Código de razão de saída</b>	<b>Tipo de saída.</b>	<b>Tipo de canal</b>
ChannelMonitoring	MQXR_INIT	Todos(as)	SDR, SVR, RCVR, RQSTR e SVRCONN, CLUSSDR e CLUSRCVR
ChannelStatistics	MQXR_INIT	Todos(as)	SDR, SVR, RCVR, RQSTR, CLUSSDR, CLUSRCVR
SharingConversations	MQXR_INIT	Todos(as)	SVRCONN, CLNTCONN
PropertyControl	MQXR_INIT	Todos(as)	SDR, SVR, CLUSSDR, CLUSRCVR

## MQCXP-Parâmetro de saída do canal

A estrutura MQCXP é transmitida para cada tipo de saída chamado por um Message Channel Agent (MCA), canal de conexão do cliente ou canal de conexão do servidor.

Consulte MQ\_CHANNEL\_EXIT.

Os campos descritos como "entrada para a saída" nas descrições a seguir são ignorados pelo canal quando a saída retorna controle para o canal. Todos os campos de entrada que a saída altera no bloco de parâmetros de saída do canal não serão preservados para sua próxima chamada. As mudanças feitas nos campos de entrada / saída (por exemplo, o campo *ExitUserArea*), são preservadas apenas para chamadas dessa instância da saída. Essas mudanças não podem ser usadas para transmitir dados entre diferentes saídas definidas no mesmo canal ou entre a mesma saída definida em canais diferentes..

### Referências relacionadas

[“Campos” na página 1574](#)

Este tópico lista todos os campos na estrutura MQCXP e descreve cada campo..

[“Declaração C” na página 1586](#)

Esta declaração é a declaração C para a estrutura MQCXP

[“Declaração COBOL” na página 1586](#)

Essa é a declaração COBOL para a estrutura MQCXP.

[“Declaração RPG \(ILE\)” na página 1587](#)

Essa é a declaração RPG para a estrutura MQCXP.

[“Declaração do assembler System/390” na página 1588](#)

Essa declaração é a declaração do assembler System/390 para a estrutura MQCXP

### Campos

Este tópico lista todos os campos na estrutura MQCXP e descreve cada campo..

*StrucId (MQCHAR4)*

Este campo especifica o identificador de estrutura.

O valor deve ser:.

## **MQCXP\_STRUC\_ID**

Identificador da estrutura do parâmetro de saída do canal.

Para a linguagem de programação C, a constante `MQCXP_STRUC_ID_ARRAY` também é definida; essa constante tem o mesmo valor que `MQCXP_STRUC_ID`, mas é uma matriz de caracteres em vez de uma cadeia.

Este é um campo de entrada para a saída

*Versão (MQLONG)*

Este campo especifica o número da versão da estrutura

O valor depende do ambiente:

### **MQCXP\_VERSION\_1**

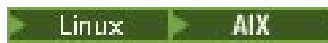
Estrutura do parâmetro de saída do canal Version-1 .

### **MQCXP\_VERSION\_2**

Estrutura do parâmetro de saída do canal Version-2 .

### **MQCXP\_VERSION\_3**

Version-3 estrutura do parâmetro de saída do canal.



O campo tem esse valor em sistemas AIX and Linux não listados em outro

lugar.

### **MQCXP\_VERSION\_4**

Estrutura do parâmetro de saída do canal Version-4

### **MQCXP\_VERSION\_5**

Estrutura do parâmetro de saída do canal Version-5

### **MQCXP\_VERSION\_6**

Version-6 estrutura do parâmetro de saída do canal.

### **MQCXP\_VERSION\_8**

Version-8 estrutura do parâmetro de saída do canal.



O campo tem esse valor em z/OS

### **MQCXP\_VERSION\_9**

Estrutura do parâmetro de saída do canal Version-9

O campo possui esse valor nos seguintes ambientes:

-  AIX
-  IBM i
-  Linux
-  Windows
-  z/OS

Os campos que existem apenas nas versões mais recentes da estrutura são identificados como tal nas descrições dos campos A constante a seguir especifica o número da versão atual:

### **MQCXP\_CURRENT\_VERSION**

Versão atual da estrutura do parâmetro de saída do canal

O valor depende do ambiente.

**Nota:** Quando uma nova versão da estrutura MQCXP é introduzida, o layout da parte existente não é alterado.. Portanto, a saída deve verificar se o número da versão é igual ou maior que a versão mais baixa que contém os campos que a saída precisa usar.

Este é um campo de entrada para a saída

#### *ExitId (MQLONG)*

Este campo especifica o tipo de saída que está sendo chamado e é definido na entrada para a rotina de saída

Os valores a seguir são possíveis:

#### **MQXT\_CHANNEL\_SEC\_EXIT**

Saída de segurança do canal

#### **MQXT\_CHANNEL\_MSG\_EXIT**

Saída da mensagem do canal

#### **MQXT\_CHANNEL\_SEND\_EXIT**

Saída de envio do canal.

#### **MQXT\_CHANNEL\_RCV\_EXIT**

Saída de recebimento do canal

#### **MQXT\_CHANNEL\_MSG\_RETRY\_EXIT**

Saída de nova tentativa de mensagem do canal.

#### **MQXT\_CHANNEL\_AUTO\_DEF\_EXIT**

Saída de definição automática do canal.

No z/OS, esse tipo de saída é suportado apenas para os canais do tipo MQCHT\_CLUSSDR e MQCHT\_CLUSRCVR

Este é um campo de entrada para a saída

#### *ExitReason (MQLONG)*

Este campo especifica o motivo pelo qual a saída está sendo chamada, e é configurada na entrada para a rotina de saída

Ele não é usado pela saída de auto-definição. Os valores a seguir são possíveis:

#### **MQXR\_INIT**

Sair da inicialização

Esse valor indica que a saída está sendo chamada pela primeira vez. Ele permite que a saída adquira e inicialize quaisquer recursos que precisar (por exemplo: memória).

#### **MQXR\_TERM**

Finalização de saída

Esse valor indica que a saída está prestes a ser finalizada. A saída deve liberar quaisquer recursos que ela adquiriu desde que foi inicializada (por exemplo: memória).

#### **MQXR\_MSG**

Processe uma mensagem

Esse valor indica que a saída está sendo invocada para processar uma mensagem. Este valor ocorre apenas para saídas de mensagens do canal.

#### **MQXR\_XMIT**

Processe uma transmissão.

Esse valor ocorre apenas para saídas de envio e recebimento do canal.

#### **MQXR\_SEC\_MSG**

Mensagem de segurança recebida

Esse valor ocorre apenas para saídas de segurança do canal.

#### **MQXR\_INIT\_SEC**

Inicie a troca de segurança

Esse valor ocorre apenas para saídas de segurança do canal.

A saída de segurança do receptor é sempre chamada com esse motivo imediatamente após ser chamado com MQXR\_INIT, para dar a oportunidade de iniciar uma troca de segurança.

Se recusar a oportunidade (retornando MQXCC\_OK em vez de MQXCC\_SEND\_SEC\_MSG ou MQXCC\_SEND\_AND\_REQUEST\_SEC\_MSG), a saída de segurança do emissor será chamada com MQXR\_INIT\_SEC.

Se a saída de segurança do receptor não iniciar uma troca de segurança (retornando MQXCC\_SEND\_SEC\_MSG ou MQXCC\_SEND\_AND\_REQUEST\_SEC\_MSG), a saída de segurança do emissor nunca será chamada com MQXR\_INIT\_SEC; em vez disso, ela será chamada com MQXR\_SEC\_MSG para processar a mensagem do receptor. (Em qualquer caso, ele é chamado primeiro com MQXR\_INIT.)

A menos que uma das saídas de segurança solicite o término do canal (configurando *ExitResponse* como MQXCC\_SUPPRESS\_FUNCTION ou MQXCC\_CLOSE\_CHANNEL), a troca de segurança deve ser concluída no lado que iniciou a troca.. Portanto, se uma saída de segurança for chamada com MQXR\_INIT\_SEC e ela iniciar uma troca, a próxima vez que a saída for chamada será com MQXR\_SEC\_MSG. Isso acontece se houver uma mensagem de segurança para a saída processar ou não. Há uma mensagem de segurança se o parceiro retornar MQXCC\_SEND\_SEC\_MSG ou MQXCF\_SEND\_AND\_REQUEST\_SEC\_MSG, mas não se o parceiro retornar MQXCC\_OK ou não houver saída de segurança no parceiro.. Se não houver nenhuma mensagem de segurança para processar, a saída de segurança na extremidade inicial será chamada novamente com um *DataLength* de zero.

### **MQXR\_RETRY**

Tente novamente uma mensagem

Esse valor ocorre apenas para saídas de nova tentativa de mensagem.

### **MQXR\_AUTO\_CLUSSDR**

Definição automática de um canal do emissor de clusters.

Esse valor ocorre apenas para saídas de definição automática de canal.

### **MQXR\_AUTO\_RECEIVER**

Definição automática de um canal receptor.

Esse valor ocorre apenas para saídas de definição automática de canal.

### **MQXR\_AUTO\_SVRCONN**

Definição automática de um canal de conexão do servidor.

Esse valor ocorre apenas para saídas de definição automática de canal.

### **MQXR\_AUTO\_CLUSRCVR**

Definição automática de um canal do receptor de clusters

Esse valor ocorre apenas para saídas de definição automática de canal.

### **MQXR\_SEC\_PARMS**

Parâmetros de segurança

Esse valor se aplica apenas a saídas de segurança e indica que uma estrutura MQCSP está sendo transmitida para a saída. Para obter mais informações, consulte [“MQCSP-Parâmetros de segurança” na página 343](#)

### **Nota:**

1. Se você tiver mais de uma saída definida para um canal, elas serão chamadas com MQXR\_INIT quando o MCA for inicializado. Além disso, elas são chamadas com MQXR\_TERM quando o MCA é finalizado.
2. Para a saída de definição automática do canal, *ExitReason* não será configurado se *Version* for menor que MQCXP\_VERSION\_4. O valor MQXR\_AUTO\_SVRCONN está implícito neste caso..

Este é um campo de entrada para a saída

*ExitResponse* (MQLONG)

Este campo especifica a resposta da saída.

Esse campo é configurado pela saída para comunicação com o MCA. Deve ser um dos valores a seguir:

## **MQXCC\_OK**

Saída concluída com êxito.

- Para a saída de segurança do canal, esse valor indica que a transferência de mensagem agora pode continuar normalmente..
- Para a saída de nova tentativa da mensagem do canal, esse valor indica que o MCA deve aguardar o intervalo de tempo retornado pela saída no campo *MsgRetryInterval* em MQCXP e, em seguida tentar a mensagem novamente...

O campo *ExitResponse2* pode conter informações adicionais

## **MQXCC\_SUPPRESS\_FUNCTION**

Suprimir função.

- Para a saída de segurança do canal, esse valor indica que o canal deve ser encerrado.
- Para a saída de mensagem do canal, esse valor indica que a mensagem não deve continuar mais para seu destino. Em vez disso, o MCA gera uma mensagem de relatório de exceção (se uma foi solicitada pelo emissor da mensagem original) e coloca a mensagem contida no buffer original na fila de mensagens não entregues (se o emissor especificou MQRO\_DEAD\_LETTER\_Q) ou a descarta (se o emissor especificou MQRO\_DISCARD\_MSG).

Para mensagens persistentes, se o emissor especificou MQRO\_DEAD\_LETTER\_Q, mas a colocação na fila de devoluções falhar ou não houver fila de devoluções, a mensagem original será deixada na fila de transmissão e a mensagem de relatório não será gerada. A mensagem original também será deixada na fila de transmissões se a mensagem de relatório não puder ser gerada com êxito

O campo *Feedback* na estrutura MQDLH no início da mensagem na fila de devoluções indica por que a mensagem foi colocada na fila de devoluções; esse código de feedback também é usado no descritor de mensagens da mensagem de relatório de exceções (se uma foi solicitada pelo emissor)...

- Para a saída de nova tentativa de mensagem do canal, este valor indica que o MCA não espera e tenta a mensagem novamente; em vez disso, o MCA continua imediatamente com seu processamento de falha normal (a mensagem é colocada na fila de mensagens não entregues ou descartada, conforme especificado pelo emissor da mensagem)...
- Para a saída de definição automática do canal, MQXCC\_OK ou MQXCC\_SUPPRESS\_FUNCTION deve ser especificado.. Se nenhum desses valores for especificado, MQXCC\_SUPPRESS\_FUNCTION será assumido por padrão e a definição automática será abandonada.

Essa resposta não é suportada para as saídas de envio e recebimento do canal.

## **MQXCC\_SEND\_SEC\_MSG**

Enviar mensagem de segurança.

Esse valor pode ser configurado apenas por uma saída de segurança do canal.. Indica que a saída forneceu uma mensagem de segurança que deve ser transmitida ao parceiro.

## **MQXCC\_SEND\_AND\_REQUEST\_SEC\_MSG**

Enviar mensagem de segurança que requer uma resposta.

Esse valor pode ser configurado apenas por uma saída de segurança do canal.. Ele indica

- que a saída forneceu uma mensagem de segurança que pode ser transmitida ao parceiro, e
- Que a saída requer uma resposta do parceiro. Se nenhuma resposta for recebida, o canal deverá ser finalizado, pois a saída ainda não decidiu se as comunicações podem continuar.

## **MQXCC\_SUPPRESS\_EXIT**

Suprimir saída.

- Esse valor pode ser configurado por todos os tipos de saída de canal diferentes de uma saída de segurança ou de uma saída de definição automática. Ele suprime qualquer chamada adicional dessa saída (como se seu nome estivesse em branco na definição de canal) até a finalização do canal, quando a saída é novamente chamada com um *ExitReason* de MQXR\_TERM.

- Se uma saída de nova tentativa de mensagem retornar esse valor, as novas tentativas de mensagens para mensagens subsequentes serão controladas pelos atributos de canal *MsgRetryCount* e *MsgRetryInterval* normalmente. Para a mensagem atual, o MCA executa o número de novas tentativas pendentes, em intervalos fornecidos pelo atributo do canal *MsgRetryInterval*, mas somente se o código de razão for aquele que o MCA normalmente tentaria novamente (consulte o campo *MsgRetryCount* descrito em “MQCD-Definição de canal” na página 1532)... O número de novas tentativas pendentes é o valor do atributo **MsgRetryCount**, menos o número de vezes que a saída retornou MQXCC\_OK para a mensagem atual; se esse número for negativo, nenhuma nova tentativa será executada pelo MCA para a mensagem atual.

### **MQXCC\_CLOSE\_CHANNEL**

Fechar canal.

Esse valor pode ser configurado por qualquer tipo de saída de canal, exceto uma saída de definição automática..

Se o compartilhamento de conversas não estiver ativado, esse valor fechará o canal.

Se as conversas de compartilhamento estiverem ativadas, esse valor terminará a conversa. Se essa conversa for a única no canal, o canal também será fechado.

Esse campo é um campo de entrada / saída da saída.

#### *ExitResponse2 (MQLONG)*

Este campo especifica a resposta secundária da saída.

Este campo é definido como zero na entrada para a rotina de saída. Ele pode ser configurado pela saída para fornecer informações adicionais para as funções do canal do IBM MQ. Ele não é usado pela saída de auto-definição.

A saída pode configurar um ou mais dos valores a seguir: Se mais de um for necessário, os valores serão adicionados. Combinações inválidas são observadas; outras combinações são permitidas.

### **MQXR2\_PUT\_WITH\_DEF\_ACTION**

Colocar com a ação padrão

Esse valor é configurado pela saída de mensagem do canal do receptor.. Indica que a mensagem deve ser colocada com a ação padrão do MCA, que é o ID do usuário padrão do MCA ou o contexto *UserIdentifier* no MQMD (descriptor de mensagens) da mensagem.

O valor é zero, que corresponde ao valor inicial configurado quando a saída é chamada. A constante é fornecida para propósitos de documentação

### **MQXR2\_PUT\_WITH\_DEF\_USERID**

Colocar com identificador de usuário padrão.

Este valor só pode ser configurado pela saída de mensagem do canal do receptor.. Ele indica que a mensagem deve ser colocada com o identificador de usuário padrão do MCA

### **MQXR2\_PUT\_WITH\_MSG\_USERID**

Colocar com o identificador de usuário da mensagem

Este valor só pode ser configurado pela saída de mensagem do canal do receptor.. Indica que a mensagem deve ser colocada com o contexto *UserIdentifier* no MQMD (descriptor de mensagem) da mensagem (isso pode ter sido modificado pela saída).

Apenas um de MQXR2\_PUT\_WITH\_DEF\_ACTION, MQXR2\_PUT\_WITH\_DEF\_USERID e MQXR2\_PUT\_WITH\_MSG\_USERID devem ser configurados

### **MQXR2\_USE\_AGENT\_BUFFER**

Use o buffer do agente

Esse valor indica que quaisquer dados a serem transmitidos estão em *AgentBuffer*, não *ExitBufferAddr*.

O valor é zero, que corresponde ao valor inicial configurado quando a saída é chamada. A constante é fornecida para propósitos de documentação

## **MQXR2\_USE\_EXIT\_BUFFER**

Utilizar buffer de saída.

Esse valor indica que quaisquer dados a serem transmitidos estão em *ExitBufferAddr*, não *AgentBuffer*.

Apenas um dos MQXR2\_USE\_AGENT\_BUFFER e MQXR2\_USE\_EXIT\_BUFFER devem ser configurados

## **MQXR2\_DEFAULT\_CONTINUATION**

Continuação padrão..

A continuação com a próxima saída na cadeia depende da resposta da última saída chamada:

- Se MQXCC\_SUPPRESS\_FUNCTION ou MQXCC\_CLOSE\_CHANNEL forem retornados, nenhuma saída adicional na cadeia será chamada
- Caso contrário, a próxima saída na cadeia será chamada..

## **MQXR2\_CONTINUE\_CHAIN**

Continue com a próxima saída

## **MQXR2\_SUPPRESS\_CHAIN**

Ignore as saídas restantes na cadeia

Este é um campo de entrada / saída na saída.

### *Feedback (MQLONG)*

Este campo especifica o código de feedback

Esse campo é configurado como MQFB\_NONE na entrada para a rotina de saída

Se uma saída de mensagem do canal configurar o campo *ExitResponse* para MQXCC\_SUPPRESS\_FUNCTION, o campo *Feedback* especificará o código de feedback que identifica por que a mensagem foi colocada na fila de mensagens não entregues e também será usado para enviar um relatório de exceção se um tiver sido solicitado. Nesse caso, se o campo *Feedback* for MQFB\_NONE, o seguinte código de feedback será usado:

## **MQFB\_STOPPED\_BY\_MSG\_EXIT**

Mensagem interrompida pela saída de mensagem do canal.

O valor retornado neste campo pelas saídas de segurança, envio, recebimento e repetição de mensagem do canal não é usado pelo MCA.

O valor retornado neste campo por saídas de definição automática não será usado se *ExitResponse* for MQXCC\_OK, mas, caso contrário, será usado para o parâmetro *AuxErrorDataInt1* na mensagem do evento...

Este é um campo de entrada / saída da saída.

### *MaxSegmentComprimento (MQLONG)*

Este campo especifica o comprimento máximo em bytes que pode ser enviado em uma única transmissão

Ele não é usado pela saída de auto-definição. Ela é de interesse para uma saída de envio de canal, porque essa saída deve assegurar que ela não aumente o tamanho de um segmento de transmissão para um valor maior que *MaxSegmentLength*. O comprimento inclui os 8 bytes iniciais que a saída não deve alterar. O valor é negociado entre as funções do canal IBM MQ quando o canal é iniciado. Consulte [Gravando programas de saída de canal](#) para obter mais informações sobre comprimentos de segmento...

O valor nesse campo não será significativo se *ExitReason* for MQXR\_INIT.

Este é um campo de entrada para a saída

### *Área ExitUser(MQBYTE16)*

Este campo especifica a área do usuário de saída-um campo disponível para a saída usar.

Ele é inicializado para zero binário antes da primeira chamada da saída (que possui um *ExitReason* configurado como MQXR\_INIT) e, posteriormente, quaisquer mudanças feitas nesse campo pela saída são preservadas nas chamadas da saída...



O seguinte valor é definido:

### **MQXUA\_NONE**

Nenhuma informação do usuário.

O valor é binário zero para o comprimento do campo.

Para a linguagem de programação C, a constante MQXUA\_NONE\_ARRAY também é definida; essa constante tem o mesmo valor que MQXUA\_NONE, mas é uma matriz de caracteres em vez de uma cadeia.

O comprimento desse campo é fornecido por MQ\_EXIT\_USER\_AREA\_LENGTH Este é um campo de entrada / saída na saída.

### *ExitData (MQCHAR32)*

Este campo especifica os dados de saída

Esse campo é configurado na entrada para a rotina de saída para informações que as funções do canal IBM MQ obtiveram da definição de canal. Se essas informações não estiverem disponíveis, este campo estará todo em branco;

O comprimento desse campo é fornecido por MQ\_EXIT\_DATA\_LENGTH

Este é um campo de entrada para a saída

Os campos a seguir nessa estrutura não estarão presentes se *Version* for menor que MQCXP\_VERSION\_2.

### *Contagem de MsgRetry(MQLONG)*

Este campo especifica o número de vezes que a mensagem foi repetida.

A primeira vez que a saída é chamada para uma mensagem específica, este campo tem o valor zero (nenhuma tentativa ainda). Em cada chamada subsequente da saída para essa mensagem, o valor é incrementado por um pelo MCA.

Este é um campo de entrada para a saída O valor nesse campo não será significativo se *ExitReason* for MQXR\_INIT. O campo não estará presente se *Version* for menor que MQCXP\_VERSION\_2.

### *Intervalo de MsgRetry(MQLONG)*

Este campo especifica o intervalo mínimo em milissegundos após o qual a operação put é tentada novamente.

A primeira vez que a saída é chamada para uma mensagem específica, esse campo contém o valor do atributo do canal *MsgRetryInterval*. A saída pode deixar o valor inalterado ou modificá-lo para especificar um intervalo de tempo diferente em milissegundos. Se a saída retornar MQXCC\_OK no *ExitResponse*, o MCA aguardará pelo menos esse intervalo de tempo antes de tentar novamente a operação MQOPEN ou MQPUT. O intervalo de tempo especificado deve ser zero ou maior.

A segunda e as vezes subsequentes em que a saída é chamada para essa mensagem, esse campo contém o valor retornado pela chamada anterior da saída.

Se o valor retornado no campo *MsgRetryInterval* for menor que zero ou maior que 999 999 999 999, e *ExitResponse* for MQXCC\_OK, o MCA ignorará o campo *MsgRetryInterval* em MQCXP e aguardará o intervalo especificado pelo atributo do canal *MsgRetryInterval*.

Este é um campo de entrada / saída na saída. O valor nesse campo não será significativo se *ExitReason* for MQXR\_INIT. O campo não estará presente se *Version* for menor que MQCXP\_VERSION\_2.

### *MsgRetryMotivo (MQLONG)*

Este campo especifica o código de razão da tentativa anterior de colocar a mensagem

Este campo é o código de razão da tentativa anterior de colocar a mensagem; ele é um dos valores MQRC\_\*.

Este é um campo de entrada para a saída O valor nesse campo não será significativo se *ExitReason* for MQXR\_INIT. O campo não estará presente se *Version* for menor que MQCXP\_VERSION\_2.

Os campos a seguir nessa estrutura não estarão presentes se *Version* for menor que MQCXP\_VERSION\_3.

#### *HeaderLength (MQLONG)*

Este campo especifica o comprimento de informações do cabeçalho

Esse campo é relevante apenas para uma saída de mensagem e uma saída de nova tentativa de mensagem. O valor é o comprimento das estruturas do cabeçalho de roteamento no início dos dados da mensagem; estas são a estrutura MQXQH, o MQMDE (cabeçalho de extensão de descrição de mensagem) e (para uma mensagem de lista de distribuição) a estrutura MQDH e as matrizes de registros MQOR e MQPMR que seguem a estrutura MQXQH.

A saída de mensagem pode examinar essas informações do cabeçalho e modificá-las se necessário, mas os dados que a saída retorna ainda devem estar no formato correto. A saída não deve, por exemplo, criptografar ou compactar os dados do cabeçalho na extremidade de envio, mesmo se a saída de mensagem na extremidade de recebimento fizer alterações de compensação.

Se a saída de mensagem modificar as informações do cabeçalho de forma a alterar seu comprimento (por exemplo, incluindo outro destino em uma mensagem da lista de distribuições), ele deverá alterar o valor de *HeaderLength* de forma correspondente antes de retornar.

Este é um campo de entrada / saída na saída. O valor nesse campo não será significativo se *ExitReason* for MQXR\_INIT. O campo não estará presente se *Version* for menor que MQCXP\_VERSION\_3.

#### *PartnerName (MQCHAR48)*

Este campo especifica o nome do parceiro

O nome do parceiro, da seguinte forma:

- Para canais SVRCONN, ele é o ID do usuário conectado no cliente.
- Para todos os outros tipos de canal, ele é o nome do gerenciador de filas do parceiro.

Quando a saída é inicializada, este campo fica em branco porque o gerenciador de filas não sabe o nome do parceiro até que a negociação inicial tenha ocorrido após...

Este é um campo de entrada para a saída. O campo não estará presente se *Version* for menor que MQCXP\_VERSION\_3.

#### *FAPLevel (MQLONG)*

Nível de Formatos Negociados e Protocolos

Este é um campo de entrada para a saída. As mudanças nesse campo devem ser feitas somente sob a direção do serviço IBM. O campo não estará presente se *Version* for menor que MQCXP\_VERSION\_3.

#### *CapabilityFlags (MQLONG)*

É possível configurar o sinalizador do recurso para MQCF\_NONE ou MQCF\_DIST\_LISTS

É possível configurar um dos seguintes sinalizadores de recurso:

##### **MQCF\_NONE**

Sem bandeiras.

##### **MQCF\_DIST\_LISTS**

Listas de distribuição suportadas.

Este é um campo de entrada para a saída. O campo não estará presente se *Version* for menor que MQCXP\_VERSION\_3.

#### *ExitNumber (MQLONG)*

Este campo especifica o número ordinal da saída.

O número ordinal da saída, dentro do tipo definido em *ExitId*. Por exemplo, se a saída que está sendo chamada for a terceira saída de mensagem definida, este campo conterá o valor 3. Se o tipo de saída for aquele para o qual uma lista de saídas não pode ser definida (por exemplo uma saída de segurança), esse campo terá o valor 1.

Este é um campo de entrada para a saída O campo não estará presente se *Version* for menor que MQCXP\_VERSION\_3.

Os seguintes campos nessa estrutura não estarão presentes se *Version* for menor que MQCXP\_VERSION\_5.

#### *ExitSpace (MQLONG)*

Este campo especifica o número de bytes no buffer de transmissão reservado para a saída usar.

Este campo é relevante apenas para uma saída de envio Ele especifica a quantidade de espaço em bytes que as funções do canal IBM MQ reservam no buffer de transmissão para a saída usar. Este campo permite que a saída adicione ao buffer de transmissão uma pequena quantidade de dados (normalmente não excedendo algumas centenas de bytes) para uso por uma saída de recebimento complementar na outra extremidade. Os dados adicionados pela saída de envio devem ser removidos pela saída de recebimento.

O valor é sempre zero no z/OS.

**Nota:** Este recurso não deve ser usado para enviar grandes quantidades de dados, pois ele pode degradar o desempenho ou mesmo inibir a operação do canal.

Configurando *ExitSpace*, a saída é garantida que sempre há pelo menos esse número de bytes disponíveis no buffer de transmissão para a saída usar. No entanto, a saída pode usar menos do que a quantidade reservada ou mais do que a quantidade reservada se houver espaço disponível no buffer de transmissão.. O espaço de saída no buffer é fornecido após os dados existentes.

*ExitSpace* pode ser configurado pela saída somente quando *ExitReason* tiver o valor MQXR\_INIT; em todos os outros casos, o valor retornado pela saída será ignorado... Na entrada para a saída, *ExitSpace* é zero para a chamada MQXR\_INIT e é o valor retornado pela chamada MQXR\_INIT em outros casos.

Se o valor retornado pela chamada MQXR\_INIT for negativo ou houver menos de 1024 bytes disponíveis no buffer de transmissão para dados de mensagem após reservar o espaço de saída solicitado para todas as saídas de envio na cadeia, o MCA emitirá uma mensagem de erro e fechará o canal. Da mesma forma, se durante a transferência de dados, as saídas na cadeia de saída de envio alocarem mais espaço do usuário do que reservaram, de forma que menos de 1024 bytes permaneçam no buffer de transmissão para dados da mensagem, o MCA gera uma mensagem de erro e fecha o canal. O limite de 1024 permite que os fluxos de controle e administrativos do canal sejam processados pela cadeia de saídas de envio, sem a necessidade de segmentação dos fluxos.

Este é um campo de entrada / saída para a saída se *ExitReason* for MQXR\_INIT, e um campo de entrada em todos os outros casos O campo não estará presente se *Version* for menor que MQCXP\_VERSION\_5.

#### *SSLCertUserId (MQCHAR12)*

Este campo especifica o UserId associado ao certificado remoto

Ele está em branco em todas as plataformas, exceto z/OS

Este é um campo de entrada para a saída O campo não estará presente se *Version* for menor que MQCXP\_VERSION\_6.

#### *SSLRemCertIssNameComprimento (MQLONG)*

Este campo especifica o comprimento em bytes do Nome Distinto completo do emissor do certificado remoto apontado por SSLCertRemoteIssuerNamePtr.

Este é um campo de entrada para a saída O campo não estará presente se *Version* for menor que MQCXP\_VERSION\_6. O valor será zero se não for um canal TLS.

#### *SSLRemCertIssNamePtr (PMQVOID)*

Este campo especifica o endereço do Nome Distinto completo do emissor do certificado remoto

Seu valor será o ponteiro nulo se não for um canal TLS.

Este é um campo de entrada para a saída O campo não estará presente se *Version* for menor que MQCXP\_VERSION\_6.

**Nota:** O comportamento das saídas de segurança do canal na determinação do Nome Distinto do Assunto e do Nome Distinto do Emissor é alterado de IBM WebSphere MQ 7.1. Para obter mais informações, consulte [Programas de saída de segurança de canal](#).

*SecurityParms (PMQCSP).*

Esse campo especifica o endereço da estrutura do MQCSP que é usada para especificar credenciais de autenticação

O valor inicial deste campo é o indicador nulo.

Este é um campo de entrada / saída na saída. O campo não estará presente se *Version* for menor que MQCXP\_VERSION\_6.

O valor que a saída retorna neste campo deve ser utilizado por IBM MQ até MQXR\_TERM.

*Compactação CurHdr(MQLONG)*

Este campo especifica qual técnica está sendo usada atualmente para compactar os dados do cabeçalho

Ele é configurado para um dos seguintes:

**MQCOMPRESS\_NONE**

Nenhuma compactação de dados do cabeçalho é executada.

**MQCOMPRESS\_SYSTEM**

A compactação de dados de cabeçalho é executada.

O valor pode ser alterado por uma saída de mensagem do canal de envio para um dos valores suportados negociados acessados a partir do campo Lista HdrCompdo MQCD. Isso permite que a técnica usada para compactar os dados do cabeçalho a serem escolhidos para cada mensagem com base no conteúdo da mensagem O valor alterado é utilizado apenas para a mensagem atual O canal será encerrado se o atributo for alterado para um valor não suportado. O valor será ignorado se alterado fora da saída de mensagem de um canal de envio.

Este é um campo de entrada / saída na saída. O campo não estará presente se *Version* for menor que MQCXP\_VERSION\_6.

*CurMsgCompactação (MQLONG)*

Este campo especifica qual técnica está sendo usada atualmente para compactar os dados da mensagem

Ele é configurado para um dos seguintes:

**MQCOMPRESS\_NONE**

Nenhuma compactação de dados do cabeçalho é executada.

**MQCOMPRESS\_RLE**

A compactação de dados da mensagem é executada usando codificação run-length.

**MQCOMPRESS\_ZLIBFAST**

A compactação dos dados da mensagem é feita usando a técnica de compactação zlib. É preferível um tempo de compactação mais rápido.

**MQCOMPRESS\_ZLIBHIGH**

A compactação dos dados da mensagem é feita usando a técnica de compactação zlib. É preferível um nível de compactação alto.

O valor pode ser alterado por uma saída de mensagem do canal de envio para um dos valores suportados negociados acessados no campo Lista MsgCompdo MQCD. Isso permite que a técnica usada para compactar os dados da mensagem seja decidida para cada mensagem com base no conteúdo da mensagem. O valor alterado é utilizado apenas para a mensagem atual O canal será encerrado se o atributo for alterado para um valor não suportado. O valor será ignorado se alterado fora da saída de mensagem de um canal de envio.

Este é um campo de entrada / saída na saída. O campo não estará presente se *Version* for menor que MQCXP\_VERSION\_6.

#### *Hconn (MQHCONN)*

Este campo especifica a manipulação da conexão que a saída usa se ela precisar fazer qualquer chamada MQI dentro da saída

Esse campo não é relevante para as saídas em execução nos canais de conexão do cliente, em que ele contém o valor MQHC\_UNUSABLE\_HCONN (-1).

Este é um campo de entrada para a saída O campo não estará presente se *Version* for menor que MQCXP\_VERSION\_7.

#### *SharingConversations (MQBOOL)*

Esse campo especifica se a conversa é a única que pode estar em execução atualmente nessa instância do canal ou se mais de uma conversa pode estar em execução atualmente nessa instância do canal...

Também indica se o programa de saída está sujeito ao risco do MQCD ser alterado por outro programa de saída em execução ao mesmo tempo.

Este campo é relevante apenas para programas de saída em execução em canais de conexão do cliente ou de conexão do servidor

Ele é configurado para um dos seguintes:

#### **FALSE**

A instância de saída é a única instância de saída que pode estar em execução atualmente nessa instância do canal... Isso permite que a saída atualize com segurança os campos MQCD sem contenção de outras saídas em execução em outras instâncias do canal. Se as alterações nos campos MQCD são atuadas pelo canal é definido pela tabela de campos MQCD em [“Mudando campos MQCD em uma saída de canal”](#) na página 1571.

#### **TRUE**

A instância de saída não é a única instância de saída que pode estar em execução atualmente nesta instância do canal... Quaisquer mudanças feitas no MQCD não são atuadas pelo canal, exceto para mudanças listadas na tabela de campos MQCD em [“Mudando campos MQCD em uma saída de canal”](#) na página 1571 para Motivos de Saída diferentes de MQXR\_INIT... Se essa saída atualizar os campos MQCD, assegure-se de que não haja contenção de outras saídas em execução em outras conversas ao mesmo tempo, fornecendo serialização entre as saídas que são executadas nessa instância do canal.

Este é um campo de entrada para a saída O campo não estará presente se *Version* for menor que MQCXP\_VERSION\_7.

#### *MCAUserSource (MQLONG)*

Este campo especifica a origem do ID do usuário MCA fornecido.

Ele pode conter um dos seguintes valores:

#### **MQUSRC\_MAP**

O ID do usuário é especificado no atributo MCAUSER

#### **MQUSRC\_CHANEXO L**

O ID do usuário é transmitido do parceiro de entrada ou especificado no campo MCAUSER definido no objeto do canal.

Este é um campo de entrada para a saída O campo não estará presente se a Versão for menor que MQCXP\_VERSION\_8..

#### *Pontos pEntry (PMQIEP)*

Este campo especifica o endereço do ponto de entrada da interface para a chamada MQI ou DCI..

O campo não estará presente se *Versão* for menor que MQCXP\_VERSION\_8.

#### *RemoteProduct (MQCHAR4)*

Este campo especifica o nome do produto remoto

Esse campo identifica o produto remoto do cliente, por exemplo, C ou Java, conforme exibido no campo **RPRODUCT** de [DISPLAY CHSATUS](#).

O campo não estará presente se *Versão* for menor que MQCXP\_VERSION\_9.

### *RemoteVersion (MQCHAR8)*

Este campo especifica o nome da versão remota

Esse campo identifica a versão das bibliotecas clientes, conforme exibido no campo **RVERSION** de DISPLAY CHSTATUS.

O campo não estará presente se *Versão* for menor que MQCXP\_VERSION\_9.

## **Declaração C**

Esta declaração é a declaração C para a estrutura MQCXP

```
typedef struct tagMQCXP MQCXP;
struct tagMQCXP {
    MQCHAR4   StrucId;           /* Structure identifier */
    MQLONG    Version;          /* Structure version number */
    MQLONG    ExitId;           /* Type of exit */
    MQLONG    ExitReason;       /* Reason for invoking exit */
    MQLONG    ExitResponse;     /* Response from exit */
    MQLONG    ExitResponse2;    /* Secondary response from exit */
    MQLONG    Feedback;         /* Feedback code */
    MQLONG    MaxSegmentLength; /* Maximum segment length */
    MQBYTE16  ExitUserArea;     /* Exit user area */
    MQCHAR32  ExitData;         /* Exit data */
    MQLONG    MsgRetryCount;    /* Number of times the message has been
    retrieved */
    MQLONG    MsgRetryInterval; /* Minimum interval in milliseconds after
    which the put operation should be
    retrieved */
    MQLONG    MsgRetryReason;   /* Reason code from previous attempt to
    put the message */
    MQLONG    HeaderLength;     /* Length of header information */
    MQCHAR48  PartnerName;      /* Partner Name */
    MQLONG    FAPLevel;         /* Negotiated Formats and Protocols
    level */
    MQLONG    CapabilityFlags;  /* Capability flags */
    MQLONG    ExitNumber;       /* Exit number */
    /* Ver:3 */
    /* Ver:4 */
    MQLONG    ExitSpace;        /* Number of bytes in transmission buffer
    reserved for exit to use */
    /* Ver:5 */
    MQCHAR12  SSLCertUserid;    /* User identifier associated
    with remote TLS certificate */
    MQLONG    SSLRemCertIssNameLength; /* Length of
    distinguished name of issuer
    of remote TLS certificate */
    MQPTR     SSLRemCertIssNamePtr; /* Address of
    distinguished name of issuer
    of remote TLS certificate */
    PMQVOID   SecurityParms;    /* Security parameters */
    MQLONG    CurHdrCompression; /* Header data compression
    used for current message */
    MQLONG    CurMsgCompression; /* Message data compression
    used for current message */
    /* Ver:6 */
    MQHCONN   Hconn;           /* Connection handle */
    MQBOOL    SharingConversations; /* Multiple conversations
    possible on channel inst? */
    /* Ver:7 */
    MQLONG    MCAUserSource;    /* Source of the provided MCA user ID */
    PMQIEP    pEntryPoints;     /* Address of the MQIEP structure */
    /* Ver:8 */
    MQCHAR4   RemoteProduct;    /* The identifier for the remote product */
    MQCHAR8   RemoteVersion;    /* The version of the remote product */
    /* Ver:9 */
};
```

## **Declaração COBOL**

Essa é a declaração COBOL para a estrutura MQCXP.

```
** MQCXP structure
```

```

10 MQCXP.
** Structure identifier
15 MQCXP-STRUCID PIC X(4).
** Structure version number
15 MQCXP-VERSION PIC S9(9) BINARY.
** Type of exit
15 MQCXP-EXITID PIC S9(9) BINARY.
** Reason for invoking exit
15 MQCXP-EXITREASON PIC S9(9) BINARY.
** Response from exit
15 MQCXP-EXITRESPONSE PIC S9(9) BINARY.
** Secondary response from exit
15 MQCXP-EXITRESPONSE2 PIC S9(9) BINARY.
** Feedback code
15 MQCXP-FEEDBACK PIC S9(9) BINARY.
** Maximum segment length
15 MQCXP-MAXSEGMENTLENGTH PIC S9(9) BINARY.
** Exit user area
15 MQCXP-EXITUSERAREA PIC X(16).
** Exit data
15 MQCXP-EXITDATA PIC X(32).
** Number of times the message has been retried
15 MQCXP-MSGRETRYCOUNT PIC S9(9) BINARY.
** Minimum interval in milliseconds after which the put operation
** should be retried
15 MQCXP-MSGRETRYINTERVAL PIC S9(9) BINARY.
** Reason code from previous attempt to put the message
15 MQCXP-MSGRETRYREASON PIC S9(9) BINARY.
** Length of header information
15 MQCXP-HEADERLENGTH PIC S9(9) BINARY.
** Partner Name
15 MQCXP-PARTNERNAME PIC X(48).
** Negotiated Formats and Protocols level
15 MQCXP-FAPLEVEL PIC S9(9) BINARY.
** Capability flags
15 MQCXP-CAPABILITYFLAGS PIC S9(9) BINARY.
** Exit number
15 MQCXP-EXITNUMBER PIC S9(9) BINARY.
** Number of bytes in transmission buffer reserved for exit to use
15 MQCXP-EXITSPACE PIC S9(9) BINARY.
** User Id associated with remote certificate
15 MQCXP-SSLCERTUSERID PIC X(12).
** Length of distinguished name of issuer of remote TLS
** certificate
15 MQCXP-SSLREMCERTISSNAMELENGTH PIC S9(9) BINARY.
** Address of distinguished name of issuer of remote TLS
** certificate
15 MQCXP-SSLREMCERTISSNAMEPTR POINTER.
** Security parameters
15 MQCXP-SECURITYPARMS PIC S9(18) BINARY.
** Header data compression used for current message
15 MQCXP-CURHDRCOMPRESSION PIC S9(9) BINARY.
** Message data compression used for current message
15 MQCXP-CURMSGCOMPRESSION PIC S9(9) BINARY.
** Connection handle
15 MQCXP-HCONN PIC S9(9) BINARY.
** Multiple conversations possible on channel instance?
15 MQCXP-SHARINGCONVERSATIONS PIC S9(9) BINARY.
** Source of the provided MCA user ID
15 MQCXP-MCAUSERSOURCE PIC S9(9) BINARY.
** Identifier of the remote product
15 MQCXP-RPRODUCT PIC X(4).
** Identifier of the remote version
15 MQCXP-RVERSION PIC X(8).

```

## Declaração RPG (ILE)

Essa é a declaração RPG para a estrutura MQCXP.

```

D*..1.....2.....3.....4.....5.....6.....7..
D* MQCXP Structure
D*
D* Structure identifier
D CXSID 1 4
D* Structure version number
D CXVER 5 8I 0
D* Type of exit
D CXXID 9 12I 0
D* Reason for invoking exit

```

```

D CXREA          13      16I 0
D* Response from exit
D CXRES          17      20I 0
D* Secondary response from exit
D CXRE2          21      24I 0
D* Feedback code
D CXFB           25      28I 0
D* Maximum segment length
D CXMSL          29      32I 0
D* Exit user area
D CXUA           33      48
D* Exit data
D CXDAT          49      80
D* Number of times the message has been retried
D CXMRC          81      84I 0
D* Minimum interval in milliseconds after which the put operation
D* should be retried
D CXMRI          85      88I 0
D* Reason code from previous attempt to put the message
D CXMRR          89      92I 0
D* Length of header information
D CXHDL          93      96I 0
D* Partner Name
D CXPNM          97      144
D* Negotiated Formats and Protocols level
D CXFAP         145      148I 0
D* Capability flags
D CXCAP         149      152I 0
D* Exit number
D CXEXN         153      156I 0
D* Number of bytes in transmission buffer reserved for exit to use
D CXHDL         157      160I 0
D* User identifier associated with remote TLS certificate
D CXSSLCU        161      172
D* Length of distinguished name of issuer of remote TLS certificate
D CXSRCINL       173      176I 0
D* Address of distinguished name of issuer of remote TLS certificate
D CXSRCINP       177      192*
D* Security parameters
D CXSECP         193      208*
D* Header data compression used for current message
D CXCHC         209      212I 0
D* Message data compression used for current message
D CXCMC         213      216I 0
D* Connection handle
D CXHCONN        217      220I 0
D* Multiple conversations possible on channel instance?
D CXSHARECONV    221      224I 0
D* Source of the provided MCA user ID
D MCAUSERSOURCE  225      228I 0
D* Identifier of the remote product
D CXRPRO         229      232I 0
D* Identifier of the remote version
D CXRVER         233      240I 0

```

## Declaração do assembler System/390

Essa declaração é a declaração do assembler System/390 para a estrutura MQCXP

```

MQCXP           DSECT
MQCXP_STRUCID   DS    CL4   Structure identifier
MQCXP_VERSION   DS    F     Structure version number
MQCXP_EXITID    DS    F     Type of exit
MQCXP_EXITREASON DS    F     Reason for invoking exit
MQCXP_EXITRESPONSE DS    F   Response from exit
MQCXP_EXITRESPONSE2 DS    F   Secondary response from exit
MQCXP_FEEDBACK  DS    F     Feedback code
MQCXP_MAXSEGMENTLENGTH DS    F   Maximum segment length
MQCXP_EXITUSERAREA DS    XL16 Exit user area
MQCXP_EXITDATA  DS    CL32   Exit data
MQCXP_MSGRETRYCOUNT DS    F   Number of times the message has been
*                               retried
MQCXP_MSGRETRYINTERVAL DS    F   Minimum interval in milliseconds
*                               after which the put operation should
*                               be retried
MQCXP_MSGRETRYREASON DS    F   Reason code from previous attempt to
*                               put the message
MQCXP_HEADERLENGTH DS    F   Length of header information
MQCXP_PARTNERNAME DS    CL48  Partner Name

```



MQCXP_FAPLEVEL	DS	F	Negotiated Formats and Protocols level
* MQCXP_CAPABILITYFLAGS	DS	F	Capability flags
MQCXP_EXITNUMBER	DS	F	Exit number
MQCXP_EXITSPEACE	DS	F	Number of bytes in transmission buffer reserved for exit to use
* MQCXP_SSLCERTUSERID	DS	CL12	User identifier associated with remote TLS certificate
* MQCXP_SSLREMCERTISSNAMELENGTH	DS	F	Length of distinguished name of issuer of remote TLS certificate
* MQCXP_SSLREMCERTISSNAMEPTR	DS	F	Address of distinguished name of issuer of remote TLS certificate
* MQCXP_SECURITYPARMS	DS	F	Address of security parameters
MQCXP_CURHDRCOMPRESSSION	DS	F	Header data compression used for current message
* MQCXP_CURMSGCOMPRESSSION	DS	F	Message data compression used for current message
* MQCXP_HCONN	DS	F	Connection handle
MQCXP_SHARINGCONVERSATIONS	DS	F	Multiple conversations possible on channel inst?
* MQCXP_MCAUSERSOURCE	DS	F	Source of the provided MCA user ID
MQCXP_RPRODUCT	DS	CL4	Identifer of the remote product
MQCXP_RVERSION	DS	CL8	Identifer of the remote version
MQCXP_LENGTH	EQU	*-MQCXP	
	ORG	MQCXP	
MQCXP_AREA	DS	CL(MQCXP_LENGTH)	

## MQXWD-Descriptor de espera de saída

A estrutura MQXWD é um parâmetro de entrada / saída na chamada MQXWAIT..

Essa estrutura é suportada apenas no z/OS

### Referências relacionadas

[“Campos” na página 1589](#)

Este tópico lista todos os campos na estrutura MQXWD e descreve cada campo..

[“Declaração C” na página 1590](#)

Esta declaração é a declaração C para a estrutura MQXWD

[“Declaração do assembler System/390” na página 1590](#)

Essa declaração é a declaração do assembler do System/390 para a estrutura MQXWD

### Campos

Este tópico lista todos os campos na estrutura MQXWD e descreve cada campo..

*StrucId (MQCHAR4)*

Este campo especifica o identificador de estrutura.

O valor deve ser:.

#### MQXWD\_STRUC\_ID

Identificador para estrutura do descritor de espera de saída.

Para a linguagem de programação C, a constante MQXWD\_STRUC\_ID\_ARRAY também é definida; essa constante tem o mesmo valor que MQXWD\_STRUC\_ID, mas é uma matriz de caracteres em vez de uma cadeia.

O valor inicial desse campo é MQXWD\_STRUC\_ID.

*Versão (MQLONG)*

Este campo especifica o número da versão da estrutura

O valor deve ser:.

#### MQXWD\_VERSION\_1

Número da versão para a estrutura do descritor de espera de saída

O valor inicial desse campo é MQXWD\_VERSION\_1.

### *Reserved1 (MQLONG)*

Este campo é reservado O seu valor deve ser zero.

Este é um campo de entrada.

### *Reserved2 (MQLONG)*

Este campo é reservado O seu valor deve ser zero.

Este é um campo de entrada.

### *Reserved3 (MQLONG)*

Este campo é reservado O seu valor deve ser zero.

Este é um campo de entrada.

### *BCE (MQLONG)*

Este campo especifica o bloco de controle de eventos para aguardar.

Este campo é o bloco de controle de eventos (ECB) a aguardar. Ele deve ser configurado como zero antes que a chamada MQXWAIT seja emitida; na conclusão bem-sucedida, ele contém o código postal.

Este é um campo de entrada / saída.

## **Declaração C**

Esta declaração é a declaração C para a estrutura MQXWD

```
typedef struct tagMQXWD MQXWD;
struct tagMQXWD {
    MQCHAR4  StrucId;    /* Structure identifier */
    MQLONG   Version;   /* Structure version number */
    MQLONG   Reserved1; /* Reserved */
    MQLONG   Reserved2; /* Reserved */
    MQLONG   Reserved3; /* Reserved */
    MQLONG   ECB;      /* Event control block to wait on */
};
```

## **Declaração do assembler System/390**

Essa declaração é a declaração do assembler do System/390 para a estrutura MQXWD

```
MQXWD          DSECT
MQXWD_STRUCID  DS    CL4  Structure identifier
MQXWD_VERSION  DS    F    Structure version number
MQXWD_RESERVED1 DS    F    Reserved
MQXWD_RESERVED2 DS    F    Reserved
MQXWD_RESERVED3 DS    F    Reserved
MQXWD_ECB      DS    F    Event control block to wait on
*
MQXWD_LENGTH   EQU    *-MQXWD
                ORG    MQXWD
MQXWD_AREA     DS    CL(MQXWD_LENGTH)
```

## **Chamada de Saída de Carga de Trabalho do Cluster e Estruturas de Dados**

Esta seção fornece informações de referência para a saída de carga do cluster e as estruturas de dados. Estas são informações de interface de programação de uso geral.

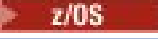
É possível gravar saídas de carga de trabalho do cluster nas linguagens de programação a seguir:

- C
- System/390 assembler ( IBM MQ for z/OS )

A chamada é descrita em:

- [“MQ\\_CLUSTER\\_WORKLOAD\\_EXIT -Descrição da chamada” na página 1591](#)

Os tipos de dados da estrutura utilizados pela saída são descritos em:

- [“MQXCLWLN -Navegar registros de carga de trabalho do Cluster” na página 1593](#)
- [“MQWXP -Estrutura do parâmetro de saída de carga do cluster” na página 1597](#)
- [“MQWDR-Estrutura de registro de destino de carga de trabalho do cluster” na página 1605](#)
- [“MQWQR -Estrutura de registro da fila de cargas de trabalho do cluster..” na página 1609](#)
- [“MQWCR -Estrutura de registro do cluster de carga de trabalho de cluster.” na página 1615](#)
-  [Comportamento assíncrono de comandos CLUSTER em z/OS](#)

Em toda esta seção, os atributos do gerenciador de fila e os atributos da fila são mostrados na íntegra. Os nomes equivalentes usados nos comandos MQSC são mostrados abaixo. Para obter detalhes de comandos MQSC, consulte [Comandos MQSC](#).

<i>Tabela 824. atributos do gerenciador de filas</i>	
<b>Nome completo</b>	<b>Nome usado no MQSC</b>
<i>ClusterWorkloadData</i>	CLWLDATA
<i>ClusterWorkloadExit</i>	CLWLEXIT
<i>ClusterWorkloadLength</i>	CLWLEN

<i>Tabela 825. Atributos da Fila</i>	
<b>Nome completo</b>	<b>Nome usado no MQSC</b>
<i>DefBind</i>	DEFBIND
<i>DefPersistence</i>	DEFPSIST
<i>DefPriority</i>	DEFPRTY
<i>InhibitPut</i>	PUT
<i>QDesc</i>	DESCR

### Tarefas relacionadas

[Gravando e Compilando Saídas de Carga de Trabalho do Cluster](#)

## MQ\_CLUSTER\_WORKLOAD\_EXIT -Descrição da chamada

A saída de carga de trabalho do cluster é chamada pelo gerenciador de filas para rotear uma mensagem para um gerenciador de filas disponível.

**Nota:** Nenhum ponto de entrada chamado MQ\_CLUSTER\_WORKLOAD\_EXIT é fornecido pelo gerenciador de filas.. Em vez disso, o nome da saída de carga de trabalho do cluster é definido para o atributo do gerenciador de filas `ClusterWorkloadExit`

A saída MQ\_CLUSTER\_WORKLOAD\_EXIT é suportada em todas as plataformas..

### Sintaxe

```
MQ_CLUSTER_WORKLOAD_EXIT (ExitParms)
```

### Referências relacionadas

[MQXCLWLN -Navegar registros de carga de trabalho do Cluster](#)

A chamada MQXCLWLN é usada para navegar pelas cadeias de registros MQWDR, MQWQR e MQWCR armazenados no cache do cluster.

[MQWXP -Estrutura do parâmetro de saída de carga do cluster](#)

A tabela a seguir resume os campos na estrutura do parâmetro de saída de carga de trabalho MQWXP-Cluster

MQWDR-Estrutura de registro de destino de carga de trabalho do cluster

A tabela a seguir resume os campos na estrutura de registro de destino de carga de trabalho do MQWDR-Cluster

MQWQR -Estrutura de registro da fila de cargas de trabalho do cluster..

A tabela a seguir resume os campos na estrutura de registro da fila de carga de trabalho do MQWQR-Cluster.

MQWCR -Estrutura de registro do cluster de carga de trabalho de cluster.

A tabela a seguir resume os campos na estrutura de registro de carga de trabalho do cluster do MQWCR

## ***Parâmetros para MQ\_CLUSTER\_WORKLOAD\_EXIT***

Descrição dos parâmetros na chamada de MQ\_CLUSTER\_WORKLOAD\_EXIT

### ***ExitParms ( MQWXP ) -entrada/saída***

Bloco de parâmetro de saída

- A saída configura as informações em MQWXP para indicar como gerenciar a carga de trabalho

### ***Referências relacionadas***

Observações de Uso

A função executada pela saída de carga de trabalho do cluster é definida pelo provedor da saída.. No entanto, a saída deve estar em conformidade com as regras definidas no bloco de controle associado MQWXP.

Chamadas de idioma para MQ\_CLUSTER\_WORKLOAD\_EXIT

O MQ\_CLUSTER\_WORKLOAD\_EXIT suporta duas linguagens, C e High Level Assembler.

### ***Observações de Uso***

A função executada pela saída de carga de trabalho do cluster é definida pelo provedor da saída.. No entanto, a saída deve estar em conformidade com as regras definidas no bloco de controle associado MQWXP.

Nenhum ponto de entrada chamado MQ\_CLUSTER\_WORKLOAD\_EXIT é fornecido pelo gerenciador de filas.. No entanto, um typedef é fornecido para o nome MQ\_CLUSTER\_WORKLOAD\_EXIT na linguagem de programação C. Use o typedef para declarar a saída gravada pelo usuário para assegurar que os parâmetros estejam corretos.

### ***Referências relacionadas***

Parâmetros para MQ\_CLUSTER\_WORKLOAD\_EXIT

Descrição dos parâmetros na chamada de MQ\_CLUSTER\_WORKLOAD\_EXIT

Chamadas de idioma para MQ\_CLUSTER\_WORKLOAD\_EXIT

O MQ\_CLUSTER\_WORKLOAD\_EXIT suporta duas linguagens, C e High Level Assembler.

## ***Chamadas de idioma para MQ\_CLUSTER\_WORKLOAD\_EXIT***

O MQ\_CLUSTER\_WORKLOAD\_EXIT suporta duas linguagens, C e High Level Assembler.

## **Chamada C**

```
MQ_CLUSTER_WORKLOAD_EXIT (&ExitParms);
```

Substitua `MQ_CLUSTER_WORKLOAD_EXIT` pelo nome da função de saída de carga de trabalho do cluster.

Declare os parâmetros **MQ\_CLUSTER\_WORKLOAD\_EXIT** da seguinte forma:

```
MQWXP ExitParms; /* Exit parameter block */
```

## Chamada do Assembler de Alto Nível

```
CALL EXITNAME,(EXITPARMS)
```

Declare os parâmetros conforme a seguir:

EXITPARMS	CMQWXP	Exit parameter block
-----------	--------	----------------------

### Referências relacionadas

[Parâmetros para MQ\\_CLUSTER\\_WORKLOAD\\_EXIT](#)

[Descrição dos parâmetros na chamada de MQ\\_CLUSTER\\_WORKLOAD\\_EXIT](#)

[Observações de Uso](#)

A função executada pela saída de carga de trabalho do cluster é definida pelo provedor da saída.. No entanto, a saída deve estar em conformidade com as regras definidas no bloco de controle associado MQWXP.

## MQXCLWLN -Navegar registros de carga de trabalho do Cluster

A chamada MQXCLWLN é usada para navegar pelas cadeias de registros MQWDR, MQWQR e MQWCR armazenados no cache do cluster.

O cache do cluster é uma área de armazenamento principal usada para armazenar informações relacionadas ao cluster..

Se o cache do cluster for estático, ele terá um tamanho fixo. Se você configurá-lo para dinâmico, o cache do cluster poderá expandir conforme necessário.

Configure o tipo do cache do cluster para STATIC ou DYNAMIC usando um parâmetro do sistema ou uma macro

- ▶ **Multi** Use o parâmetro do sistema `ClusterCacheType` on [Multiplataformas](#).
- ▶ **z/OS** Use o parâmetro `CLCACHE` na macro `CSQ6SYSP` em z/OS

## Sintaxe

```
MQXCLWLN (ExitParms, CurrentRecord, NextOffset, NextRecord, Compcode, Reason)
```

### Referências relacionadas

[MQ\\_CLUSTER\\_WORKLOAD\\_EXIT -Descrição da chamada](#)

A saída de carga de trabalho do cluster é chamada pelo gerenciador de filas para rotear uma mensagem para um gerenciador de filas disponível

[MQWXP -Estrutura do parâmetro de saída de carga do cluster](#)

A tabela a seguir resume os campos na estrutura do parâmetro de saída de carga de trabalho MQWXP -Cluster

[MQWDR-Estrutura de registro de destino de carga de trabalho do cluster](#)

A tabela a seguir resume os campos na estrutura de registro de destino de carga de trabalho do MQWDR -Cluster

[MQWQR -Estrutura de registro da fila de cargas de trabalho do cluster..](#)

A tabela a seguir resume os campos na estrutura de registro da fila de carga de trabalho do MQWQR -Cluster.

[MQWCR -Estrutura de registro do cluster de carga de trabalho de cluster.](#)

A tabela a seguir resume os campos na estrutura de registro de carga de trabalho do cluster do MQWCR

## **Parâmetros para MQXCLWLN -Navegar registros de carga de trabalho do Cluster**

Descrição dos parâmetros na chamada de MQXCLWLN

### **ExitParms ( MQWXP ) -entrada/saída**

Bloco de parâmetro de saída

Essa estrutura contém informações relacionadas à invocação da saída. A saída configura informações nessa estrutura para indicar como gerenciar a carga de trabalho..

### **CurrentRecord ( MQPTR ) -entrada**

Endereço do registro atual..

Essa estrutura contém informações relacionadas ao endereço do registro atualmente sendo examinado pela saída. O registro deve ser um dos seguintes tipos:

- Registro de destino de cargas de trabalho do cluster ( MQWDR ).
- Registro da fila de carga de trabalho do cluster ( MQWQR )
- Registro de cluster de carga de trabalho do cluster ( MQWCR ).

### **NextOffset ( MQLONG ) -entrada**

Compensação do próximo registro.

Esta estrutura contém informações relacionadas ao deslocamentos do próximo registro ou estrutura. *NextOffset* é o valor do campo de deslocamento apropriado no registro atual e deve ser um dos seguintes campos:

- Campo ChannelDefoffset em MQWDR
- Campo ClusterRecde ClusterRec em MQWDR
- Campo ClusterRecDeslocamento em MQWQR
- Campo ClusterRecDeslocamento em MQWCR

### **NextRecord ( MQPTR ) -saída**

Endereço do próximo registro ou estrutura.

Essa estrutura contém informações relacionadas ao endereço do próximo registro ou estrutura. Se *CurrentRecord* for o endereço de um MQWDR e *NextOffset* for o valor do campo ChannelDefOffset , *NextRecord* será o endereço da estrutura de definição de canal ( MQCD ).

Se não houver nenhum próximo registro ou estrutura, o gerenciador de filas configura *NextRecord* para o ponteiro nulo e a chamada retorna o código de conclusão MQCC\_WARNING e o código de razão MQRC\_NO\_RECORD\_AVAILABLE.

### **CompCode ( MQLONG ) -saída**

Código de conclusão.

O código de conclusão possui um dos seguintes valores:

#### **MQCC\_OK**

Indica conclusão bem-sucedida.

#### **MQCC\_WARNING**

Aviso (conclusão parcial).

#### **MQCC\_FAILED**

A chamada falhou.

### **Motivo ( MQLONG ) -saída**

Código de razão qualificando CompCode

Se CompCode for MQCC\_OK:

#### **MQRC\_NONE**

( 0, X'0000' )

Não há motivo para reportar.

Se *CompCode* for MQCC\_WARNING:

#### **MQRC\_NO\_RECORD\_AVAILABLE**

**(2359, X'0937')**

Nenhum registro disponível. Uma chamada MQXCLWLN foi emitida a partir de uma saída de carga de trabalho do cluster para obter o endereço do próximo registro na cadeia O registro atual é o último registro na cadeia. Ação corretiva: Nenhuma.

Se *CompCode* for MQCC\_FAILED:

#### **MQRC\_CURRENT\_RECORD\_ERROR**

**(2357, X'0935')**

O parâmetro **CurrentRecord** não é válido Uma chamada MQXCLWLN foi emitida a partir de uma saída de carga de trabalho do cluster para obter o endereço do próximo registro na cadeia O endereço especificado pelo parâmetro **CurrentRecord** não é o endereço de um registro válido **CurrentRecord** deve ser o endereço de um registro de destino, MQWDR, registro de filas (MQWQR) ou registro de cluster (MQWCR) residentes no cache do cluster. Ação corretiva: Assegure que a saída de carga de trabalho do cluster passe o endereço de um registro válido que reside no cache do cluster.

#### **MQRC\_ENVIRONMENT\_ERROR**

**(2012, X'07DC')**

A chamada não é válida no ambiente Uma chamada MQXCLWLN foi emitida, mas não de uma saída de carga de trabalho do cluster.

#### **MQRC\_NEXT\_OFFSET\_ERROR**

**(2358, X'0936')**

O parâmetro **NextOffset** não é válido Uma chamada MQXCLWLN foi emitida a partir de uma saída de carga de trabalho do cluster para obter o endereço do próximo registro na cadeia O deslocamento especificado pelo parâmetro **NextOffset** não é válido **NextOffset** deve ser o valor de um dos seguintes campos:

- Campo `ChannelDefoffset` em MQWDR
- Campo `ClusterRecde ClusterRec` em MQWDR
- Campo `ClusterRecDeslocamento` em MQWQR
- Campo `ClusterRecDeslocamento` em MQWCR

Ação corretiva: Assegure que o valor especificado para o parâmetro **NextOffset** seja o valor de um dos campos listados anteriormente.

#### **MQRC\_NEXT\_RECORD\_ERROR**

**(2361, X'0939')**

O parâmetro **NextRecord** não é válido

#### **MQRC\_WXP\_ERROR**

**(2356, X'0934')**

A estrutura do parâmetro de saída da carga de trabalho não é válida Uma chamada MQXCLWLN foi emitida a partir de uma saída de carga de trabalho do cluster para obter o endereço do próximo registro na cadeia A estrutura do parâmetro de saída da carga de trabalho **ExitParms** não é válida, por uma das seguintes razões:

- O ponteiro de parâmetro não é válido. Nem sempre é possível detectar ponteiros de parâmetros inválidos; se não forem detectados, ocorrerão resultados imprevisíveis.
- O campo `StrucId` não é MQWXP\_STRUC\_ID..
- O campo `Versão` não é MQWXP\_VERSION\_2..
- O campo `Contexto` não contém o valor transmitido para a saída pelo gerenciador de filas.

Ação corretiva: Assegure que o parâmetro especificado para **ExitParms** seja a estrutura MQWXP que foi transmitida para a saída quando a saída foi chamada.

#### **Referências relacionadas**

[Notas de uso para MQXCLWLN-Navegar registros de carga de trabalho do Cluster](#)

Use MQXCLWLN para navegar através de registros de cluster, mesmo se o cache for estático

Chamadas de linguagem de MQXCLWLN

O MQXCLWLN suporta duas linguagens, C e High Level Assembler.

### **Notas de uso para MQXCLWLN-Navegar registros de carga de trabalho do Cluster**

Use MQXCLWLN para navegar através de registros de cluster, mesmo se o cache for estático

Se o cache do cluster for dinâmico, a chamada MQXCLWLN deverá ser usada para navegar pelos registros. A saída termina de forma anormal se a aritmética simples de ponteiro e deslocamento for usada para navegar pelos registros..

Se o cache do cluster for estático, o MQXCLWLN não precisará ser usado para navegar pelos registros. Geralmente, use MQXCLWLN mesmo quando o cache for estático. É possível então alterar o cache do cluster para ser dinâmico sem precisar alterar a saída de carga de trabalho.

#### **Referências relacionadas**

[Parâmetros para MQXCLWLN -Navegar registros de carga de trabalho do Cluster](#)

[Descrição dos parâmetros na chamada de MQXCLWLN](#)

Chamadas de linguagem de MQXCLWLN

O MQXCLWLN suporta duas linguagens, C e High Level Assembler.

### **Chamadas de linguagem de MQXCLWLN**

O MQXCLWLN suporta duas linguagens, C e High Level Assembler.

## **Chamada C**

```
MQXCLWLN (&ExitParms, CurrentRecord, NextOffset, &NextRecord, &CompCode, &Reason) ;
```

Declare os parâmetros conforme a seguir:

```
typedef struct tagMQXCLWLN {
MQWXP   ExitParms;      /* Exit parameter block */
MQPTR   CurrentRecord; /* Address of current record*/
MQLONG  NextOffset;    /* Offset of next record */
MQPTR   NextRecord;    /* Address of next record or structure */
MQLONG  CompCode;      /* Completion code */
MQLONG  Reason;        /* Reason code qualifying CompCode */
}
```

## **Chamada do Assembler de Alto Nível**

```
CALL MQXCLWLN, (CLWLEXITPARMS, CURRENTRECORD, NEXTOFFSET, NEXTRECORD, COMPCODE, REASON)
```

Declare os parâmetros conforme a seguir:

```
CLWLEXITPARMS CMQWXP, Cluster workload exit parameter block
CURRENTRECORD CMQWDRA, Current record
NEXTOFFSET    DS F      Next offset
NEXTRECORD    DS F      Next record
COMPCODE      DS F      Completion code
REASON        DS F      Reason code qualifying COMPCODE
```

#### **Referências relacionadas**

[Parâmetros para MQXCLWLN -Navegar registros de carga de trabalho do Cluster](#)

[Descrição dos parâmetros na chamada de MQXCLWLN](#)

[Notas de uso para MQXCLWLN-Navegar registros de carga de trabalho do Cluster](#)

Use MQXCLWLN para navegar através de registros de cluster, mesmo se o cache for estático



## MQWXP -Estrutura do parâmetro de saída de carga do cluster

A tabela a seguir resume os campos na estrutura do parâmetro de saída de carga de trabalho MQWXP -Cluster

<i>Tabela 826. Campos em MQWXP</i>		
<b>Campo</b>	<b>Descrição</b>	<b>Página</b>
<i>StrucId</i>	Identificador de estruturação	<a href="#">StrucId</a>
<i>Version</i>	Número de versão da estrutura	<a href="#">Versão</a>
<i>ExitId</i>	Tipo de saída	<a href="#">ExitId</a>
<i>ExitReason</i>	Motivo para chamar a saída.	<a href="#">ExitReason</a>
<i>ExitResponse</i>	Resposta da saída	<a href="#">ExitResponse</a>
<i>ExitResponse2</i>	Resposta secundária da saída	<a href="#">ExitResponse2</a>
<i>Feedback</i>	Código de feedback	<a href="#">Feedback</a>
<i>Flags</i>	Sinaliza valores. Esses sinalizadores de bits são usados para indicar informações sobre a mensagem que está sendo colocada	<a href="#">Sinalizadores</a>
<i>ExitUserArea</i>	Sair da área do usuário..	<a href="#">ExitUserArea</a>
<i>ExitData</i>	Dados de Saída	<a href="#">ExitData</a>
<i>MsgDescPtr</i>	Endereço do descritor de mensagem ( MQMD )	<a href="#">MsgDescPtr</a>
<i>MsgBufferPtr</i>	Endereço do buffer contendo alguns ou todos os dados da mensagem	<a href="#">MsgBufferPtr</a>
<i>MsgBufferLength</i>	Comprimento do buffer contendo dados da mensagem	<a href="#">MsgBufferComprimento</a>
<i>MsgLength</i>	Comprimento da mensagem completa	<a href="#">MsgLength</a>
<i>QName</i>	Nome da fila	<a href="#">QName</a>
<i>QMgrName</i>	Nome do gerenciador de filas locais	<a href="#">QMgrName</a>
<i>DestinationCount</i>	Número de destinos possíveis	<a href="#">DestinationCount</a>
<i>DestinationChosen</i>	Destino escolhido	<a href="#">DestinationChosen</a>
<i>DestinationArrayPtr</i>	Endereço de uma matriz de ponteiros para registros de destino ( MQWDR )	<a href="#">DestinationArrayPtr</a>
<i>QArrayPtr</i>	Endereço de uma matriz de ponteiros para registros de fila ( MQWQR )	<a href="#">QArrayPtr</a>
<b>Nota:</b> Os campos restantes serão ignorados se a Versão for menor que MQWXP_VERSION_2		
<i>CacheContext</i>	Informações de contexto	<a href="#">CacheContext</a>
<i>CacheType</i>	Tipo de cache do cluster	<a href="#">CacheType</a>
<b>Nota:</b> Os campos restantes serão ignorados se a Versão for menor que MQWXP_VERSION_3		
<i>CLWLMRUChannels</i>	Número máximo de canais de cluster de saída ativos permitidos..	<a href="#">CLWLMRUChannels</a>
<b>Nota:</b> Os campos restantes serão ignorados se a Versão for menor que MQWXP_VERSION_4		

Tabela 826. Campos em MQWXP (continuação)		
Campo	Descrição	Página
<code>pEntryPoints</code>	Endereço da estrutura MQIEP para permitir que chamadas MQI e DCI sejam feitas	<a href="#">pEntryPontos</a>

A estrutura do parâmetro de saída de carga de trabalho do cluster descreve as informações transmitidas para a saída de carga de trabalho do cluster.

A estrutura do parâmetro de saída da carga de trabalho do cluster é suportada em todas as plataformas

Além disso, as estruturas MQWXP1, MQWXP2 e MQWXP3 estão disponíveis para compatibilidade com versões anteriores.

### Referências relacionadas

[MQ\\_CLUSTER\\_WORKLOAD\\_EXIT](#) -Descrição da chamada

A saída de carga de trabalho do cluster é chamada pelo gerenciador de filas para rotear uma mensagem para um gerenciador de filas disponível

[MQXCLWLN](#) -Navegar registros de carga de trabalho do Cluster

A chamada MQXCLWLN é usada para navegar pelas cadeias de registros MQWDR, MQWQR e MQWCR armazenados no cache do cluster.

[MQWDR](#)-Estrutura de registro de destino de carga de trabalho do cluster

A tabela a seguir resume os campos na estrutura de registro de destino de carga de trabalho do MQWDR -Cluster

[MQWQR](#) -Estrutura de registro da fila de cargas de trabalho do cluster..

A tabela a seguir resume os campos na estrutura de registro da fila de carga de trabalho do MQWQR -Cluster.

[MQWCR](#) -Estrutura de registro do cluster de carga de trabalho de cluster.

A tabela a seguir resume os campos na estrutura de registro de carga de trabalho do cluster do MQWCR

### **Campos em MQWXP -Estrutura do parâmetro de saída de carga do cluster**

Descrição dos campos na estrutura do parâmetro de saída de carga de trabalho MQWXP -Cluster

#### **StrucId (MQCHAR4)-entrada**

O identificador de estrutura para a estrutura do parâmetro de saída de carga de trabalho do cluster.

- O valor StrucId é MQWXP\_STRUC\_ID.
- Para a linguagem de programação C, a constante MQWXP\_STRUC\_ID\_ARRAY também é definida.. Ele tem o mesmo valor que MQWXP\_STRUC\_ID É uma matriz de caracteres em vez de uma sequência.

#### **Versão (MQLONG)-entrada de**

Indica o número da versão da estrutura Versão usa um dos seguintes valores:

##### **MQWXP\_VERSION\_1**

Version-1 estrutura do parâmetro de saída de carga de trabalho do cluster.

MQWXP\_VERSION\_1 é suportado em todos os ambientes.

##### **MQWXP\_VERSION\_2**

Estrutura do parâmetro de saída da carga de trabalho do cluster Version-2 .

MQWXP\_VERSION\_2 é suportado nos ambientes a seguir:

-  AIX
-  IBM i
-  Linux
-  Windows

### **MQWXP\_VERSION\_3**

Estrutura do parâmetro de saída da carga de trabalho do cluster Version-3

MQWXP\_VERSION\_3 é suportado nos ambientes a seguir:

-  AIX
-  IBM i
-  Linux
-  Windows

### **MQWXP\_VERSION\_4**

Estrutura do parâmetro de saída de carga de trabalho do cluster Version-4

MQWXP\_VERSION\_4 é suportado nos ambientes a seguir:

-  AIX
-  IBM i
-  Linux
-  Windows

### **MQWXP\_CURRENT\_VERSION**

Versão atual da estrutura do parâmetro de saída de carga de trabalho do cluster

### **ExitId (MQLONG)-entrada**

Indica o tipo de saída que está sendo chamado A saída de carga de trabalho do cluster é a única saída suportada

- O valor ExitId deve ser MQXT\_CLUSTER\_WORKLOAD\_EXIT

### **ExitReason (MQLONG)-entrada**

Indica o motivo para chamar a saída de carga do cluster. ExitReason usa um dos seguintes valores:

#### **MQXR\_INIT**

Indica que a saída está sendo invocada pela primeira vez

Adquirir e inicializar quaisquer recursos que a saída possa precisar, como armazenamento principal.

#### **MQXR\_TERM**

Indica que a saída está prestes a ser finalizada

Libere todos os recursos que a saída pode ter adquirido desde que foi inicializada, como armazenamento principal

#### **MQXR\_CLWL\_OPEN**

Chamado por MQOPEN

#### **MQXR\_CLWL\_PUT**

Chamado por MQPUT ou MQPUT1

#### **MQXR\_CLWL\_MOVE**

Chamado pelo MCA quando o estado do canal foi alterado.

#### **MQXR\_CLWL\_REPOS**

Chamado por MQPUT ou MQPUT1 para uma mensagem PCF do gerenciador de repositório

#### **MQXR\_CLWL\_REPOS\_MOVE**

Chamado pelo MCA para uma mensagem PCF do gerenciador de repositório se o estado do canal foi alterado.

### **ExitResponse (MQLONG)-saída**

Configure ExitResponse para indicar se o processamento da mensagem continua.. Deve ser um dos valores a seguir:

**MQXCC\_OK**

Continue processando a mensagem normalmente

- `DestinationChosen` identifica o destino para o qual a mensagem deve ser enviada.

**MQXCC\_SUPPRESS\_FUNCTION**

Descontinue o processamento da mensagem

- As ações executadas pelo gerenciador de filas dependem do motivo pelo qual a saída foi chamada::

<i>Tabela 827. Ações tomadas pelo gerenciador de filas</i>	
<b>ExitReason</b>	<b>Ações realizadas</b>
<ul style="list-style-type: none"> <li>– MQXR_CLWL_OPEN</li> <li>– MQXR_CLWL_REPOS</li> <li>– MQXR_CLWL_PUT</li> </ul>	MQOPEN, MQPUT ou MQPUT1 falha de chamada com código de conclusão MQCC_FAILED e código de razão MQRC_STOPPED_BY_CLUSTER_EXIT.
<ul style="list-style-type: none"> <li>– MQXR_CLWL_MOVE</li> <li>– MQXR_CLWL_REPOS_MOVE</li> </ul>	A mensagem é colocada na fila de mensagens não entregues

**MQXCC\_SUPPRESS\_EXIT**

Continue processando a mensagem atual normalmente Não chame a saída novamente até que o gerenciador de filas seja encerrado.

O gerenciador de filas processa mensagens subsequentes como se o atributo do gerenciador de filas `ClusterWorkloadExit` estivesse em branco. `DestinationChosen` identifica o destino para o qual a mensagem atual é enviada.

**Qualquer outro valor**

Processe a mensagem como se `MQXCC_SUPPRESS_FUNCTION` fosse especificado

**ExitResponse2 (MQLONG)-entrada/saída**

Configure `ExitResponse2` para fornecer mais informações ao gerenciador de filas.

- `MQXR2_STATIC_CACHE` é o valor padrão e é configurado na entrada para a saída..
- Quando `ExitReason` tiver o valor `MQXR_INIT`, a saída poderá configurar um dos seguintes valores em `ExitResponse2`:

**MQXR2\_STATIC\_CACHE**

A saída requer um cache de cluster estático

- Se o cache do cluster for estático, a saída não precisará usar a chamada `MQXCLWLN` para navegar pelas cadeias de registros no cache do cluster..
- Se o cache do cluster for dinâmico, a saída não poderá navegar corretamente pelos registros no cache.

**Nota:** O gerenciador de filas processa o retorno da chamada `MQXR_INIT` como se a saída tivesse retornado `MQXCC_SUPPRESS_EXIT` no campo `ExitResponse` .

**MQXR2\_DYNAMIC\_CACHE**

A saída pode operar com um cache estático ou dinâmico

- Se a saída retornar esse valor, a saída deverá usar a chamada `MQXCLWLN` para navegar nas cadeias de registros no cache do cluster...

**Feedback (MQLONG)-entrada**

Um campo reservado.. O valor é zero.

**Sinalizadores (MQLONG)-entrada**

Indica informações sobre a mensagem sendo colocada.

- O valor de `Flags` é `MQWXP_PUT_BY_CLUSTER_CHL`. A mensagem é originada de um canal de cluster, em vez de localmente ou de um canal não cluster. Em outras palavras, a mensagem foi proveniente de outro gerenciador de filas do cluster

#### **Reservado (MQLONG)-entrada**

Um campo reservado.. O valor é zero.

#### **ExitUserArea (MQBYTE16)-entrada/saída**

Configure `ExitUserArea` para se comunicar entre chamadas para a saída.

- `ExitUserArea` é inicializado para binário zero antes da primeira chamada da saída. Quaisquer mudanças feitas nesse campo pela saída são preservadas nas chamadas da saída que ocorrem entre a chamada `MQCONN` e a chamada `MQDISC` correspondente. O campo é reconfigurado para binário zero quando a chamada `MQDISC` ocorre.
- A primeira chamada da saída é indicada pelo campo `ExitReason` com o valor `MQXR_INIT`.
- As seguintes constantes são definidas:

##### **MQXUA\_NONE -sequência**

##### **MQXUA\_NONE\_ARRAY -matriz de caractere**

Nenhuma informação do usuário. Ambas constantes são binárias zero para o comprimento do campo.

##### **MQ\_EXIT\_USER\_AREA\_LENGTH**

O comprimento da área `ExitUser`.

#### **ExitData (MQCHAR32)-entrada**

O valor do atributo do gerenciador de filas `ClusterWorkloadData`. Se nenhum valor tiver sido definido para esse atributo, este campo conterá apenas espaços em branco.

- O comprimento de `ExitData` é fornecido por `MQ_EXIT_DATA_LENGTH`

#### **MsgDescPtr (PMQMD)-entrada**

O endereço de uma cópia do descritor de mensagem (`MQMD`) para a mensagem sendo processada.

- Quaisquer mudanças feitas no descritor de mensagem pela saída são ignoradas pelo gerenciador de fila
- Se `ExitReason` tiver um dos seguintes valores `MsgDescPtr` for configurado para o ponteiro nulo e nenhum descritor de mensagens for transmitido para a saída:
  - `MQXR_INIT`
  - `MQXR_TERM`
  - `MQXR_CLWL_OPEN`

#### **MsgBufferPtr (PMQVOID)-entrada**

O endereço de um buffer contendo uma cópia dos primeiros bytes de `MsgBufferComprimento` dos dados da mensagem.

- Quaisquer mudanças feitas nos dados da mensagem pela saída são ignoradas pelo gerenciador de fila.
- Nenhum dado de mensagem é passado para a saída quando:
  - `MsgDescPtr` é o ponteiro nulo.
  - A mensagem não possui dados..
  - O atributo do gerenciador de filas `ClusterWorkloadLength` é zero.
 Nesses casos, `MsgBufferPtr` é o ponteiro nulo.

#### **MsgBufferComprimento (MQLONG)-entrada**

O comprimento do buffer que contém os dados da mensagem transmitidos para a saída

- O comprimento é controlado pelo atributo do gerenciador de filas `ClusterWorkloadLength` ..
- O comprimento pode ser inferior ao comprimento da mensagem completa, consulte `MsgLength`.

**MsgLength (MQLONG)-entrada**

O comprimento da mensagem completa transmitida para a saída

- `MsgBufferLength` pode ser inferior ao comprimento da mensagem completa.
- `MsgLength` será zero se `ExitReason` for `MQXR_INIT`, `MQXR_TERM` ou `MQXR_CLWL_OPEN`.

**QName (MQCHAR48)-entrada**

O nome da fila de destino. A fila é uma fila de clusters

- O comprimento de `QName` é `MQ_Q_NAME_LENGTH`.

**QMgrName (MQCHAR48)-entrada**

O nome do gerenciador de filas locais que chamou a saída de carga de trabalho do cluster

- O comprimento de `QMgrName` é `MQ_Q_MGR_NAME_LENGTH`.

**DestinationCount (MQLONG)-entrada**

O número de destinos possíveis Os destinos são instâncias da fila de destino e são descritos por registros de destino

- Um registro de destino é uma estrutura de `MQWDR` Há uma estrutura para cada rota possível para cada instância de fila.
- `MQWDR` estruturas são endereçadas por uma matriz de ponteiros, consulte `DestinationArrayPtr`.

**DestinationChosen (MQLONG)-entrada / saída**

O destino escolhido..

- O número da estrutura `MQWDR` que identifica a rota e a instância da fila na qual a mensagem deve ser enviada.
- O valor está no intervalo 1- `DestinationCount`.
- Na entrada para a saída, `DestinationChosen` indica a rota e a instância da fila que o gerenciador de filas selecionou. A saída pode aceitar essa opção ou escolher uma rota e instância de fila diferentes.
- O valor configurado pela saída deve estar no intervalo de 1 a `DestinationCount` Se qualquer outro valor for retornado, o gerenciador de filas usará o valor de `DestinationChosen` na saída.

**DestinationArrayPtr (PPMQWDR)-entrada**

O endereço de uma matriz de ponteiros para registros de destino (`MQWDR`).

- Há registros de destino `DestinationCount` .

**QArrayPtr (PPMQQR)-entrada**

O endereço de uma matriz de ponteiros para registros de fila (`MQWQR`).

- Se registros de fila estiverem disponíveis, haverá `DestinationCount` deles.
- Se nenhum registro de fila estiver disponível, `QArrayPtr` será o ponteiro nulo.

**Nota:** `QArrayPtr` pode ser o ponteiro nulo mesmo quando `DestinationCount` é maior que zero.

**CacheContext (MQPTR): Versão 2-entrada**

O campo `CacheContext` é reservado para uso pelo gerenciador de filas.. A saída não deve alterar o valor desse campo

**CacheType (MQLONG): Versão 2-entrada**

O cache do cluster possui um dos seguintes tipos:

**MQCLCT\_STATIC**

O cache é estático

- O tamanho do cache é fixo e não pode crescer conforme o gerenciador de filas opera.
- Não é necessário usar a chamada `MQXCLWLN` para navegar pelos registros nesse tipo de cache

**MQCLCT\_DYNAMIC**

O cache é dinâmico

- O tamanho do cache pode aumentar para acomodar as informações de cluster variáveis.
- Deve-se usar a chamada MQXCLWLN para navegar pelos registros nesse tipo de cache

#### **CLWLMRUChannels (MQLONG): Versão 3-entrada**

Indica o número máximo de canais de cluster de saída ativos, a serem considerados para uso pelo algoritmo de opção de carga de trabalho do cluster

- CLWLMRUChannels é um valor de 1 a 999 999 999.

#### **pEntryPoints (PMQIEP): Versão 4**

O endereço de uma estrutura MQIEP através da qual chamadas MQI e DCI podem ser feitas.

#### **Referências relacionadas**

Valores iniciais e declarações de idioma para MQWXP

Valores iniciais e declarações de linguagem C e High Level Assembler para MQWXP -Estrutura do parâmetro de saída de carga de trabalho do cluster.

#### **Valores iniciais e declarações de idioma para MQWXP**

Valores iniciais e declarações de linguagem C e High Level Assembler para MQWXP -Estrutura do parâmetro de saída de carga de trabalho do cluster.

<i>Tabela 828. Campos em MQWXP</i>		
<b>Nome de Campo</b>	<b>Nome da constante</b>	<b>Valor da constante</b>
<i>StrucId</i>	MQWXP_STRUC_ID	'WXP'
<i>Version</i>	MQWXP_VERSION_2	2
<i>ExitId</i>	Nenhum	0
<i>ExitReason</i>	MQXCC_OK	0
<i>ExitResponse</i>	Nenhum	0
<i>ExitResponse2</i>	Nenhum	0
<i>Flags</i>	Nenhum	0
<i>ExitUserArea</i>	{MQXUA_NONE_ARRAY}	0
<i>ExitData</i>	Nenhum	" "
<i>MsgDescPtr</i>	Nenhum	NULL
<i>MsgBufferPtr</i>	Nenhum	NULL
<i>MsgBufferLength</i>	Nenhum	0
<i>MsgBufferPtr</i>	Nenhum	0
<i>QName</i>	Nenhum	" "
<i>QMgrName</i>	Nenhum	" "
<i>DestinationCount</i>	Nenhum	0
<i>DestinationChosen</i>	Nenhum	0
<i>DestinationArrayPtr</i>	Nenhum	NULL
<i>QArrayPtr</i>	Nenhum	NULL
<i>CacheContext</i>	Nenhum	NULL
<i>CacheType</i>	MQCLCT_DYNAMIC	1
<i>CLWLMRUChannels</i>	Nenhum	0

Tabela 828. Campos em MQWXP (continuação)

Nome de Campo	Nome da constante	Valor da constante
<i>pEntryPoints</i>	Nenhum	NULL

**Notas:**

1. O símbolo ~ representa um único caractere em branco.
2. Na linguagem de programação C, a variável macro MQWXP\_DEFAULT contém os valores padrão. Use-o da seguinte maneira para fornecer valores iniciais para os campos na estrutura:

```
MQWDR MyWXP = {MQWXP_DEFAULT};
```

## Declaração C

```
typedef struct tagMQWXP {
    MQCHAR4   StrucId;           /* Structure identifier */
    MQLONG    Version;         /* Structure version number */
    MQLONG    ExitId;          /* Type of exit */
    MQLONG    ExitReason;      /* Reason for invoking exit */
    MQLONG    ExitResponse;    /* Response from exit */
    MQLONG    ExitResponse2;   /* Reserved */
    MQLONG    Feedback;        /* Reserved */
    MQLONG    Flags;           /* Flags */
    MQBYTE16  ExitUserArea;    /* Exit user area */
    MQCHAR32  ExitData;        /* Exit data */
    PMQMD     MsgDescPtr;      /* Address of message descriptor */
    PMQVOID   MsgBufferPtr;    /* Address of buffer containing some
                               or all of the message data */
    MQLONG    MsgBufferLength; /* Length of buffer containing message
                               data */
    MQLONG    MsgLength;       /* Length of complete message */
    MQCHAR48  QName;           /* Queue name */
    MQCHAR48  QMgrName;        /* Name of local queue manager */
    MQLONG    DestinationCount; /* Number of possible destinations */
    MQLONG    DestinationChosen; /* Destination chosen */
    PPMQWDR   DestinationArrayPtr; /* Address of an array of pointers to
                               destination records */
    PPMQWQR   QArrayPtr;      /* Address of an array of pointers to
                               queue records */

    /* version 1 */
    MQPTR     CacheContext;    /* Context information */
    MQLONG    CacheType;       /* Type of cluster cache */
    /* version 2 */
    MQLONG    CLWLMRChannels; /* Maximum number of most recently
                               used cluster channels */

    /* version 3 */
    PMQIEP    pEntryPoints;    /* Address of the MQIEP structure */
    /* version 4 */
};
```

## High Level Assembler

```
MQWXP          DSECT
MQWXP_STRUCID  DS    CL4      Structure identifier
MQWXP_VERSION  DS     F       Structure version number
MQWXP_EXITID   DS     F       Type of exit
MQWXP_EXITREASON DS    F      Reason for invoking exit
MQWXP_EXITRESPONSE DS    F    Response from exit
MQWXP_EXITRESPONSE2 DS    F    Reserved
MQWXP_FEEDBACK DS     F       Reserved
MQWXP_RESERVED DS     F       Reserved
MQWXP_EXITUSERAREA DS   XL16   Exit user area
MQWXP_EXITDATA DS    CL32     Exit data
MQWXP_MSGDESCPTR DS     F      Address of message
*              descriptor
MQWXP_MSGBUFFERPTR DS    F     Address of buffer containing
*              some or all of the message
```



* MQWXP_MSGBUFFERLENGTH	DS	F	data	Length of buffer containing message data
* MQWXP_MSGLENGTH	DS	F		Length of complete message
* MQWXP_QNAME	DS	CL48		Queue name
* MQWXP_QMGRNAME	DS	CL48		Name of local queue manager
* MQWXP_DESTINATIONCOUNT	DS	F		Number of possible destinations
* MQWXP_DESTINATIONCHOSEN	DS	F		Destination chosen
* MQWXP_DESTINATIONARRAYPTR	DS	F		Address of an array of pointers to destination records
* MQWXP_QARRAYPTR	DS	F		Address of an array of pointers to queue records
* MQWXP_CACHECONTEXT	DS	F		Context information
* MQWXP_CACHETYPE	DS	F		Type of cluster cache
* MQWXP_CLWLMRCHANNELS	DS	F		Number of most recently used channels for workload balancing
* MQWXP_LENGTH	EQU	*-MQWXP		Length of structure
	ORG	MQWXP		
MQWXP_AREA	DS	CL(MQWXP_LENGTH)		

### Referências relacionadas

Campos em MQWXP -Estrutura do parâmetro de saída de carga do cluster

Descrição dos campos na estrutura do parâmetro de saída de carga de trabalho MQWXP -Cluster

### MQWDR-Estrutura de registro de destino de carga de trabalho do cluster

A tabela a seguir resume os campos na estrutura de registro de destino de carga de trabalho do MQWDR -Cluster

Tabela 829. Campos em MQWDR		
Campo	Descrição	Página
<i>StrucId</i>	Identificador de estruturação	<a href="#">StrucId</a>
<i>Version</i>	Número de versão da estrutura	<a href="#">Versão</a>
<i>StrucLength</i>	Comprimento da estrutura MQWDR	<a href="#">StrucLength</a>
<i>QMgrFlags</i>	Sinalizadores do gerenciador de filas	<a href="#">QMgrFlags</a>
<i>QMgrIdentifier</i>	Identificador do Gerenciador de Filas	<a href="#">QMgrIdentifier</a>
<i>QMgrName</i>	Nome do gerenciador de filas	<a href="#">QMgrName</a>
<i>ClusterRecOffset</i>	Compensação lógica do primeiro registro de cluster (MQWCR)	<a href="#">ClusterRecDeslocamento</a>
<i>ChannelState</i>	Estado do canal	<a href="#">ChannelState</a>
<i>ChannelDefOffset</i>	Deslocamento lógico da estrutura de definição de canal (MQCD)	<a href="#">ChannelDefDeslocamento</a>
<b>Nota:</b> Os campos restantes serão ignorados se a Versão for menor que MQWDR_VERSION_2		
<i>DestSeqNumber</i>	Número de sequência de destino do canal	<a href="#">DestSeqDestSeq</a>
<i>DestSeqFactor</i>	Fator de sequência de destino do canal para ponderação	<a href="#">DestSeqDestSeq</a>

A estrutura do registro de destino da carga de trabalho do cluster contém informações relacionadas a um dos destinos possíveis para a mensagem. Há uma estrutura de registro de destino de carga do cluster para cada instância da fila de destino.

A estrutura de registro de destino de carga do cluster é suportada em todos os ambientes.

Além disso, as estruturas MQWDR1 e MQWDR2 estão disponíveis para compatibilidade com versões anteriores.

### **Referências relacionadas**

MQ\_CLUSTER\_WORKLOAD\_EXIT -Descrição da chamada

A saída de carga de trabalho do cluster é chamada pelo gerenciador de filas para rotear uma mensagem para um gerenciador de filas disponível

MQXCLWLN -Navegar registros de carga de trabalho do Cluster

A chamada MQXCLWLN é usada para navegar pelas cadeias de registros MQWDR, MQWQRe MQWCR armazenados no cache do cluster.

MQWXP -Estrutura do parâmetro de saída de carga do cluster

A tabela a seguir resume os campos na estrutura do parâmetro de saída de carga de trabalho MQWXP -Cluster

MQWQR -Estrutura de registro da fila de cargas de trabalho do cluster..

A tabela a seguir resume os campos na estrutura de registro da fila de carga de trabalho do MQWQR -Cluster.

MQWCR -Estrutura de registro do cluster de carga de trabalho de cluster.

A tabela a seguir resume os campos na estrutura de registro de carga de trabalho do cluster do MQWCR

### ***Campos em MQWDR-Estrutura de registro de destino de carga de trabalho do cluster***

Descrição dos parâmetros na estrutura de registro de destino de carga de trabalho do cluster do MQWDR

#### **StrucId ( MQCHAR4 ) -entrada**

O identificador da estrutura para a estrutura do registro de destino da carga de trabalho do cluster..

- O valor StrucId é MQWDR\_STRUC\_ID.
- Para a linguagem de programação C, a constante MQWDR\_STRUC\_ID\_ARRAY também é definida.. Ele tem o mesmo valor que MQWDR\_STRUC\_ID É uma matriz de caracteres em vez de uma sequência.

#### **Versão ( MQLONG ) -entrada**

O número da versão da estrutura Versão usa um dos seguintes valores:

##### **MQWDR\_VERSION\_1**

Registro de destino da carga de trabalho do cluster Version-1 .

##### **MQWDR\_VERSION\_2**

Registro de destino da carga de trabalho do cluster Version-2 .

##### **MQWDR\_CURRENT\_VERSION**

Versão atual do registro de destino da carga de trabalho do cluster

#### **StrucLength ( MQLONG ) -entrada**

O comprimento da estrutura MQWDR . StrucLength usa um dos seguintes valores:

##### **MQWDR\_LENGTH\_1**

Comprimento do registro de destino de carga de trabalho do cluster version-1

##### **MQWDR\_LENGTH\_2**

Comprimento do registro de destino de carga de trabalho do cluster version-2

##### **MQWDR\_CURRENT\_LENGTH**

Comprimento da versão atual do registro de destino da carga de trabalho do cluster

#### **QMgrFlags ( MQLONG ) -entrada**

Sinalizadores do gerenciador de filas indicando propriedades do gerenciador de filas que hospeda a instância da fila de destino descrita pela estrutura MQWDR . Os sinalizadores a seguir são definidos:

##### **MQQMF\_REPOSITORY\_Q\_MGR**

O destino é um gerenciador de filas de repositório completo.

##### **MQQMF\_CLUSSDR\_USER\_DEFINED**

O canal do emissor de clusters foi definido manualmente.

**MQQMF\_CLUSSDR\_AUTO\_DEFINED**

O canal do emissor de clusters foi definido automaticamente.

**MQQMF\_AVAILABLE**

Gerenciador de filas de destino está disponível para receber mensagens.

**Outros valores**

Outros sinalizadores no campo podem ser configurados pelo gerenciador de filas para propósitos internos

**QMgrIdentifier (MQCHAR48) -entrada**

O identificador do gerenciador de filas é um identificador exclusivo para o gerenciador de filas que hospeda a instância da fila de destino descrita pela estrutura MQWDR .

- O identificador é gerado pelo gerenciador de filas.
- O comprimento de QMgrIdentifier é MQ\_Q\_MGR\_IDENTIFIER\_LENGTH

**QMgrName (MQCHAR48) -entrada**

O nome do gerenciador de filas que hospeda a instância de fila de destino descrita pela estrutura MQWDR

- QMgrName pode ser o nome do gerenciador de filas locais, bem como outro gerenciador de filas no cluster.
- O comprimento de QMgrName é MQ\_Q\_MGR\_NAME\_LENGTH.

**ClusterRecDeslocamento (MQLONG) -entrada**

O deslocamento lógico da primeira estrutura MQWCR que pertence à estrutura MQWDR .

- Para caches estáticos, ClusterRecOffset é o deslocamento da primeira estrutura MQWCR que pertence à estrutura MQWDR .
- O deslocamento é medido em bytes do início da estrutura MQWDR .
- Não utilize o deslocamento lógico para a aritmética de ponteiro com caches dinâmicos Para obter o endereço do próximo registro, a chamada MQXCLWLN deve ser usada..

**ChannelState (MQLONG) -entrada**

O estado do canal que vincula o gerenciador de filas locais ao gerenciador de filas identificado pela estrutura MQWDR . Os valores a seguir são possíveis:

**MQCHS\_BINDING**

O canal está negociando com o parceiro.

**MQCHS\_INACTIVE**

O canal não está ativo.

**MQCHS\_INITIALIZING**

O canal está inicializando.

**MQCHS\_PAUSED**

O canal foi pausado

**MQCHS\_REQUESTING**

Canal do solicitante está solicitando conexão.

**MQCHS\_RETRYING**

O canal está tentando novamente estabelecer conexão.

**MQCHS\_RUNNING**

O canal está sendo transferido ou aguardando mensagens.

**MQCHS\_STARTING**

O canal está aguardando para se tornar ativo.

**MQCHS\_STOPPING**

O canal está parando.

**MQCHS\_STOPPED**

O canal foi interrompido.

### ChannelDefDeslocamento ( MQLONG ) -entrada

O deslocamento lógico da definição de canal ( MQCD ) para o canal que vincula o gerenciador de fila local ao gerenciador de fila identificado pela estrutura MQWDR .

- ChannelDefOffset é como ClusterRecOffset
- O deslocamento lógico não pode ser usado na aritmética de ponteiro Para obter o endereço do próximo registro, a chamada MQXCLWLN deve ser usada..

### DestSeqFator ( MQLONG ) -entrada

O fator de sequência de destino que permite uma escolha do canal com base no peso

- DestSeqFactor é usado antes que o gerenciador de filas o altere.
- O gerenciador de carga de trabalho aumenta o DestSeqFactor de uma maneira que assegura que as mensagens sejam distribuídas em canais inativos de acordo com seu peso.

### DestSeqNúmero ( MQLONG ) -entrada

O valor de destino do canal do cluster antes que o gerenciador de filas o altere

- O gerenciador de carga de trabalho aumenta DestSeqNumber toda vez que uma mensagem é colocada nesse canal.
- As saídas de carga de trabalho podem usar DestSeqNumber para decidir qual canal colocar uma mensagem para baixo.

### Referências relacionadas

Valores iniciais e declarações de idioma para MQWDR

Valores iniciais e declarações de linguagem C e High Level Assembler para MQWDR -Registro de destino de carga de trabalho do cluster

### Valores iniciais e declarações de idioma para MQWDR

Valores iniciais e declarações de linguagem C e High Level Assembler para MQWDR -Registro de destino de carga de trabalho do cluster

Tabela 830. Campos em MQWDR		
Nome de Campo	Nome da constante	Valor da constante
<i>StrucId</i>	MQWDR_STRUC_ID	'WDR→'
<i>Version</i>	MQWDR_VERSION_1	1
<i>StrucLength</i>	MQWDR_CURRENT_LENGTH <sup>3</sup>	136
<i>QMgrFlags</i>	MQWDR_NONE	0
<i>QMgrIdentifier</i>	Nenhum	" "
<i>QMgrName</i>	Nenhum	" "
<i>ClusterRecOffset</i>	Nenhum	0
<i>ChannelState</i>	Nenhum	0
<i>ChannelDefOffset</i>	Nenhum	0
<i>DestSeqNumber</i>	Nenhum	0
<i>DestSeqFactor</i>	Nenhum	0

Tabela 830. Campos em MQWDR (continuação)

Nome de Campo	Nome da constante	Valor da constante
<b>Notas:</b>		
1. O símbolo <code>\_</code> representa um único caractere em branco.		
2. Na linguagem de programação C, a variável macro <code>MQWDR_DEFAULT</code> contém os valores padrão. Use-o da seguinte maneira para fornecer valores iniciais para os campos na estrutura:		
<pre>MQWDR MyWDR = {MQWDR_DEFAULT};</pre>		
3. Os valores iniciais configuram intencionalmente o comprimento da estrutura para o comprimento da versão atual e não a versão 1 da estrutura.		

## High Level Assembler

```
MQWDR                DSECT
MQWDR_STRUCID        DS    CL4      Structure identifier
MQWDR_VERSION        DS    F        Structure version number
MQWDR_STRUCLNGTH     DS    F        Length of MQWDR structure
MQWDR_QMGRFLAGS      DS    F        Queue manager flags
MQWDR_QMGRIDENTIFIER DS    CL48     Queue manager identifier
MQWDR_QMGRNAME       DS    CL48     Queue manager name
MQWDR_CLUSTERRECOFFSET DS    F        Offset of first cluster
*                    record
MQWDR_CHANNELSTATE   DS    F        Channel state
MQWDR_CHANNELDEFOFFSET DS    F        Offset of channel definition
*                    structure
MQWDR_LENGTH         EQU    *-MQWDR Length of structure
MQWDR_AREA           DS    CL(MQWDR_LENGTH)
ORG    MQWDR
```

## Declaração C

```
typedef struct tagMQWDR {
    MQCHAR4    StrucId;           /* Structure identifier */
    MQLONG     Version;          /* Structure version number */
    MQLONG     StrucLength;      /* Length of MQWDR structure */
    MQLONG     QMgrFlags;        /* Queue managerflags */
    MQCHAR48   QMgrIdentifier;   /* Queue manageridentifier */
    MQCHAR48   QMgrName;        /* Queue manager name */
    MQLONG     ClusterRecOffset; /* Offset of first cluster record */
    MQLONG     ChannelState;     /* Channel state */
    MQLONG     ChannelDefOffset; /* Offset of channel definition structure */
    /* Ver:1 */
    MQLONG     DestSeqNumber;    /* Cluster channel destination sequence number */
    MQINT64    DestSeqFactor;    /* Cluster channel factor sequence number */
    /* Ver:2 */
};
```

### Referências relacionadas

Campos em MQWDR-Estrutura de registro de destino de carga de trabalho do cluster

Descrição dos parâmetros na estrutura de registro de destino de carga de trabalho do cluster do MQWDR

## MQWQR -Estrutura de registro da fila de cargas de trabalho do cluster..

A tabela a seguir resume os campos na estrutura de registro da fila de carga de trabalho do MQWQR -Cluster.

<i>Tabela 831. Campos em MQWQR</i>		
<b>Campo</b>	<b>Descrição</b>	<b>Página</b>
<i>StrucId</i>	Identificador de estruturação	<a href="#">StrucId</a>
<i>Version</i>	Número de versão da estrutura	<a href="#">Versão</a>
<i>StrucLength</i>	Comprimento da estrutura MQWQR	<a href="#">StrucLength</a>
<i>QFlags</i>	Sinalizações de fila	<a href="#">QFlags</a>
<i>QName</i>	Nome da fila	<a href="#">QName</a>
<i>QMgrIdentifier</i>	Identificador do Gerenciador de Filas	<a href="#">QMgrIdentifier</a>
<i>ClusterRecOffset</i>	Deslocamento do primeiro registro de cluster (MQWCR).	<a href="#">ClusterRecDeslocamento</a>
<i>QType</i>	Tipo de fila	<a href="#">QType</a>
<i>QDesc</i>	Descrição da fila	<a href="#">QDesc</a>
<i>DefBind</i>	Ligação padrão	<a href="#">DefBind</a>
<i>DefPersistence</i>	Persistência padrão de mensagem	<a href="#">DefPersistence</a>
<i>DefPriority</i>	Prioridade padrão da mensagem	<a href="#">DefPriority</a>
<i>InhibitPut</i>	Se as operações put na fila são permitidas	<a href="#">InhibitPut</a>
<b>Nota:</b> Os campos restantes serão ignorados se a Versão for menor que MQWQR_VERSION_2		
<i>CWLQueuePriority</i>	Um valor de 0 a 9 representando a prioridade da fila	<a href="#">CLWLQueuePriority</a>
<i>CLWLQueueRank</i>	Um valor de 0 a 9 que representa a classificação da fila	<a href="#">CLWLQueueRank</a>
<b>Nota:</b> Os campos restantes serão ignorados se a Versão for menor que MQWQR_VERSION_3		
<i>DefPutResponse</i>	Resposta Put Padrão	<a href="#">DefPutResposta</a>

A estrutura de registro da fila de carga de trabalho do cluster contém informações relacionadas a um dos destinos possíveis para a mensagem. Há uma estrutura de registro da fila de cargas de trabalho do cluster para cada instância da fila de destino.

A estrutura de registro da fila de carga de trabalho do cluster é suportada em todos os ambientes.

Além disso, as estruturas MQWQR1 e MQWQR2 estão disponíveis para compatibilidade com versões anteriores.

### Referências relacionadas

[MQ\\_CLUSTER\\_WORKLOAD\\_EXIT](#) -Descrição da chamada

A saída de carga de trabalho do cluster é chamada pelo gerenciador de filas para rotear uma mensagem para um gerenciador de filas disponível.

[MQXCLWLN](#) -Navegar registros de carga de trabalho do Cluster

A chamada MQXCLWLN é usada para navegar pelas cadeias de registros MQWDR, MQWQRe MQWCR armazenados no cache do cluster.

[MQWXP](#) -Estrutura do parâmetro de saída de carga do cluster

A tabela a seguir resume os campos na estrutura do parâmetro de saída de carga de trabalho MQWXP -Cluster.

[MQWDR](#)-Estrutura de registro de destino de carga de trabalho do cluster

A tabela a seguir resume os campos na estrutura de registro de destino de carga de trabalho do MQWDR -Cluster.

MQWCR -Estrutura de registro do cluster de carga de trabalho de cluster.

A tabela a seguir resume os campos na estrutura de registro de carga de trabalho do cluster do MQWCR

### **Campos em MQWQR -Estrutura de registro da fila de carga de trabalho do cluster.**

Descrição dos campos na estrutura de registro da fila de carga de trabalho do cluster do MQWQR

#### **StrucId ( MQCHAR4 ) -entrada**

O identificador de estrutura para a estrutura de registro da fila de carga de trabalho do cluster..

- O valor StrucId é MQWQR\_STRUC\_ID.
- Para a linguagem de programação C, a constante MQWQR\_STRUC\_ID\_ARRAY também é definida. Ele tem o mesmo valor que MQWQR\_STRUC\_ID É uma matriz de caracteres em vez de uma sequência.

#### **Versão ( MQLONG ) -entrada**

O número da versão da estrutura Versão usa um dos seguintes valores:

##### **MQWQR\_VERSION\_1**

Registro da fila de carga de trabalho do cluster Version-1 .

##### **MQWQR\_VERSION\_2**

Registro da fila de carga de trabalho do cluster Version-2 .

##### **MQWQR\_VERSION\_3**

Registro da fila de carga de trabalho do cluster Version-3 .

##### **MQWQR\_CURRENT\_VERSION**

Versão atual do registro da Fila de Carga de Trabalho do Cluster

#### **StrucLength ( MQLONG ) -entrada**

O comprimento da estrutura MQWQR . StrucLength usa um dos seguintes valores:

##### **MQWQR\_LENGTH\_1**

Comprimento do registro da fila de cargas de trabalho do cluster version-1

##### **MQWQR\_LENGTH\_2**

Comprimento do registro de fila de carga de trabalho do cluster version-2 .

##### **MQWQR\_LENGTH\_3**

Comprimento do registro da Fila de Carga de Trabalho do Cluster version-3

##### **MQWQR\_CURRENT\_LENGTH**

Comprimento da versão atual do registro de fila de carga de trabalho do cluster

#### **QFlags ( MQLONG ) -entrada**

As sinalizações de fila indicam propriedades da fila. Os sinalizadores a seguir são definidos:

##### **MQQF\_LOCAL\_Q**

O destino é uma fila local

##### **MQQF\_CLWL\_USEQ\_ANY**

Permitir uso de filas locais e remotas em puts.

##### **MQQF\_CLWL\_USEQ\_LOCAL**

Permitir apenas fila local.

##### **Outros valores**

Outros sinalizadores no campo podem ser configurados pelo gerenciador de filas para propósitos internos

#### **QName ( MQCHAR48 ) -entrada**

O nome da fila que é um dos destinos possíveis da mensagem..

- O comprimento de QName é MQ\_Q\_NAME\_LENGTH.

#### **QMgrIdentifier ( MQCHAR48 ) -entrada**

O identificador do gerenciador de filas é um identificador exclusivo para o gerenciador de filas que hospeda a instância da fila descrita pela estrutura MQWQR .

- O identificador é gerado pelo gerenciador de filas.

- O comprimento de `QMgrIdentifier` é `MQ_Q_MGR_IDENTIFIER_LENGTH`

#### **ClusterRecDeslocamento ( MQLONG ) -entrada**

O deslocamento lógico da primeira estrutura `MQWCR` que pertence à estrutura `MQWQR`.

- Para caches estáticos, `ClusterRecOffset` é o deslocamento da primeira estrutura `MQWCR` pertencente à estrutura `MQWQR`.
- O deslocamento é medido em bytes do início da estrutura `MQWQR`.
- Não utilize o deslocamento lógico para a aritmética de ponteiro com caches dinâmicos. Para obter o endereço do próximo registro, a chamada `MQXCLWLN` deve ser usada..

#### **QType ( MQLONG ) -entrada**

O tipo de fila da fila de destino. Os valores a seguir são possíveis:

##### **MQCQT\_LOCAL\_Q**

Fila local.

##### **MQCQT\_ALIAS\_Q**

Fila de alias.

##### **MQCQT\_REMOTE\_Q**

Fila remota.

##### **MQCQT\_Q\_MGR\_ALIAS**

Alias do gerenciador de filas..

#### **QDesc ( MQCHAR64 ) -entrada**

O atributo da fila de descrição de filas definido no gerenciador de filas que hospeda a instância da fila de destino descrita na estrutura `MQWQR`.

- O comprimento de `QDesc` é `MQ_Q_DESC_LENGTH`

#### **DefBind ( MQLONG -entrada**

O atributo da fila de ligação padrão definido no gerenciador de filas que hospeda a instância da fila de destino descrita na estrutura `MQWQR`. `MQBND_BIND_ON_OPEN` ou `MQBND_BIND_ON_GROUP` deve ser especificado ao usar grupos com clusters. Os seguintes valores são possíveis:

##### **MQBND\_BIND\_ON\_OPEN**

Ligação corrigida pela chamada `MQOPEN`.

##### **MQBND\_BIND\_NOT\_FIXED**

Ligação não corrigida.

##### **MQBND\_BIND\_ON\_GROUP**

Permite que um aplicativo solicite que um grupo de mensagens seja alocado na mesma instância de destino.

#### **DefPersistence ( MQLONG ) -entrada**

O atributo da fila de persistência de mensagem padrão definido no gerenciador de fila que hospeda a instância da fila de destino descrita pela estrutura `MQWQR`. Os valores a seguir são possíveis:

##### **MQPER\_PERSISTENT**

A mensagem é persistente

##### **MQPER\_NOT\_PERSISTENT**

A mensagem não é persistente

#### **DefPriority ( MQLONG -entrada**

O atributo da fila de prioridade de mensagens padrão definido no gerenciador de fila que hospeda a instância da fila de destino descrita na estrutura `MQWQR`. O intervalo de prioridade é 0- `MaxPriority`.

- 0 é a prioridade mais baixa.
- `MaxPriority` é o atributo do gerenciador de fila do gerenciador de filas que hospeda essa instância da fila de destino



### **InhibitPut ( MQLONG ) -entrada**

O atributo da fila de entrada inibida definido no gerenciador de filas que hospeda a instância da fila de destino descrita pela estrutura MQWQR Os valores a seguir são possíveis:

#### **MQQA\_PUT\_INHIBITED**

Operações put são inibidas.

#### **MQQA\_PUT\_ALLOWED**

Operações put são permitidas.

### **CLWLQueuePriority ( MQLONG ) -entrada**

O atributo de prioridade da fila de carga de trabalho do cluster definido no gerenciador de filas que hospeda a instância da fila de destino descrita na estrutura MQWQR

### **CLWLQueueRank ( MQLONG -entrada**

A classificação da fila de carga do cluster definida no gerenciador de filas que hospeda a instância da fila de destino descrita pela estrutura MQWQR .

### **DefPutDefPut ( MQLONG ). -entrada**

O atributo da fila de resposta de colocação padrão definido no gerenciador de filas que hospeda a instância da fila de destino descrita pela estrutura MQWQR Os valores a seguir são possíveis:

#### **MQPRT\_SYNC\_RESPONSE**

Resposta síncrona para chamadas de MQPUT ou MQPUT1

#### **MQPRT\_ASYNC\_RESPONSE**

Resposta assíncrona para chamadas de MQPUT ou MQPUT1

### **Referências relacionadas**

Valores iniciais e declarações de idioma para MQWQR

Valores iniciais e declarações de linguagem C e High Level Assembler para MQWQR -registro de fila de carga de trabalho do cluster..

### **Valores iniciais e declarações de idioma para MQWQR**

Valores iniciais e declarações de linguagem C e High Level Assembler para MQWQR -registro de fila de carga de trabalho do cluster..

<i>Tabela 832. Campos em MQWQR</i>		
<b>Nome de Campo</b>	<b>Nome da constante</b>	<b>Valor da constante</b>
<i>StrucId</i>	MQWQR_STRUC_ID_ARRAY	'WQR→'
<i>Version</i>	MQWQR_VERSION_1	1
<i>StrucLength</i>	MQWQR_CURRENT_LENGTH <sup>3</sup>	212
<i>QFlags</i>	Nenhum	0
<i>QName</i>	Nenhum	" "
<i>QMgrIdentifier</i>	Nenhum	" "
<i>ClusterRecOffset</i>	Nenhum	0
<i>QType</i>	Nenhum	0
<i>QDesc</i>	Nenhum	" "
<i>DefBind</i>	Nenhum	0
<i>DefPersistence</i>	Nenhum	0
<i>DefPriority</i>	Nenhum	0
<i>InhibitPut</i>	Nenhum	0
<i>CLWLQueuePriority</i>	Nenhum	0

Tabela 832. Campos em MQWQR (continuação)

Nome de Campo	Nome da constante	Valor da constante
<i>CLWLQueueRank</i>	Nenhum	0
<i>DefPutResponse</i>	Nenhum	1

**Notas:**

1. O símbolo ~ representa um único caractere em branco.
2. Na linguagem de programação C, a variável macro MQWQR\_DEFAULT contém os valores padrão. Use-o da seguinte maneira para fornecer valores iniciais para os campos na estrutura:

```
MQWQR MyWQR = {MQWQR_DEFAULT};
```

3. Os valores iniciais configuram intencionalmente o comprimento da estrutura para o comprimento da versão atual e não a versão 1 da estrutura.

**Declaração C**

```
typedef struct tagMQWQR {
    MQCHAR4   StrucId;           /* Structure identifier */
    MQLONG    Version;          /* Structure version number */
    MQLONG    StrucLength;      /* Length of MQWQR structure */
    MQLONG    QFlags;          /* Queue flags */
    MQCHAR48  QName;           /* Queue name */
    MQCHAR48  QMgrIdentifier;    /* Queue manager identifier */
    MQLONG    ClusterRecOffset; /* Offset of first cluster record */
    MQLONG    QType;           /* Queue type */
    MQCHAR64  QDesc;           /* Queue description */
    MQLONG    DefBind;         /* Default binding */
    MQLONG    DefPersistence;   /* Default message persistence */
    MQLONG    DefPriority;      /* Default message priority */
    MQLONG    InhibitPut;      /* Whether put operations on the queue
                               are allowed */

    /* version 2 */
    MQLONG    CLWLQueuePriority; /* Queue priority */
    MQLONG    CLWLQueueRank;    /* Queue rank */
    /* version 3 */
    MQLONG    DefPutResponse;   /* Default put response */
};
```

**High Level Assembler**

```
MQWQR          DSECT
MQWQR_STRUCID  DS    CL4      Structure identifier
MQWQR_VERSION  DS     F       Structure version number
MQWQR_STRUCLNGTH DS    F       Length of MQWQR structure
MQWQR_QFLAGS   DS     F       Queue flags
MQWQR_QNAME    DS   CL48     Queue name
MQWQR_QMGRIDENTIFIER DS CL48  Queue manager identifier
MQWQR_CLUSTERRECOFFSET DS    F  Offset of first cluster
*              record
MQWQR_QTYPE    DS     F       Queue type
MQWQR_QDESC    DS   CL64     Queue description
MQWQR_DEFBIND  DS     F       Default binding
MQWQR_DEFPERSISTENCE DS    F  Default message persistence
MQWQR_DEFPRIORITY DS    F  Default message priority
MQWQR_INHIBITPUT DS    F  Whether put operations on
*              the queue are allowed
MQWQR_DEFPUTRESPONSE DS    F  Default put response
MQWQR_LENGTH   EQU  *-MQWQR  Length of structure
                ORG    MQWQR
MQWQR_AREA     DS    CL(MQWQR_LENGTH)
```

## Referências relacionadas

[Campos em MQWQR -Estrutura de registro da fila de carga de trabalho do cluster.](#)

Descrição dos campos na estrutura de registro da fila de carga de trabalho do cluster do MQWQR

## MQWCR -Estrutura de registro do cluster de carga de trabalho de cluster.

A tabela a seguir resume os campos na estrutura de registro de carga de trabalho do cluster do MQWCR

Campo	Descrição	Página
<i>ClusterName</i>	Nome do cluster	<a href="#">ClusterName</a>
<i>ClusterRecOffset</i>	Deslocamento do próximo registro do cluster (MQWCR)	<a href="#">ClusterRecDeslocamento</a>
<i>ClusterFlags</i>	Sinalizações de cluster	<a href="#">ClusterFlags</a>

A estrutura de registro do cluster de cargas de trabalho do cluster contém informações sobre um cluster. Para cada cluster ao qual a fila de destino pertence, há uma estrutura de registros de clusters de carga de trabalho do cluster.

A estrutura de registro do cluster de carga de trabalho do cluster é suportada em todos os ambientes

## Referências relacionadas

[MQ\\_CLUSTER\\_WORKLOAD\\_EXIT -Descrição da chamada](#)

A saída de carga de trabalho do cluster é chamada pelo gerenciador de filas para rotear uma mensagem para um gerenciador de filas disponível

[MQXCLWLN -Navegar registros de carga de trabalho do Cluster](#)

A chamada MQXCLWLN é usada para navegar pelas cadeias de registros MQWDR, MQWQR e MQWCR armazenados no cache do cluster.

[MQWXP -Estrutura do parâmetro de saída de carga do cluster](#)

A tabela a seguir resume os campos na estrutura do parâmetro de saída de carga de trabalho MQWXP -Cluster

[MQWDR-Estrutura de registro de destino de carga de trabalho do cluster](#)

A tabela a seguir resume os campos na estrutura de registro de destino de carga de trabalho do MQWDR -Cluster

[MQWQR -Estrutura de registro da fila de cargas de trabalho do cluster..](#)

A tabela a seguir resume os campos na estrutura de registro da fila de carga de trabalho do MQWQR -Cluster.

## **Campos na estrutura de registro do cluster de carga de trabalho MQWCR -Cluster.**

Descrição dos campos na estrutura de registro do cluster de carga de trabalho MQWCR -Cluster.

### **ClusterName ( MQCHAR48 ) -entrada**

O nome de um cluster ao qual a instância da fila de destino que possui a estrutura do MQWCR pertence. A instância da fila de destino é descrita pela estrutura MQWDR .

- O comprimento de `ClusterName` é `MQ_CLUSTER_NAME_LENGTH`.

### **ClusterRecDeslocamento ( MQLONG ) -entrada**

A compensação lógica da próxima estrutura MQWCR .

- Se não houver mais estruturas MQWCR , `ClusterRecOffset` será zero.
- O deslocamento é medido em bytes desde o início da estrutura MQWCR .

### **ClusterFlags ( MQLONG ) -entrada**

Os sinalizadores de cluster indicam as propriedades do gerenciador de filas identificado pela estrutura MQWCR. Os sinalizadores a seguir são definidos:

### **MQQMF\_REPOSITORY\_Q\_MGR**

O destino é um gerenciador de filas de repositório completo.

### **MQQMF\_CLUSSDR\_USER\_DEFINED**

O canal do emissor de clusters foi definido manualmente.

### **MQQMF\_CLUSSDR\_AUTO\_DEFINED**

O canal do emissor de clusters foi definido automaticamente.

### **MQQMF\_AVAILABLE**

Gerenciador de filas de destino está disponível para receber mensagens.

### **Outros valores**

Outros sinalizadores no campo podem ser configurados pelo gerenciador de filas para propósitos internos

## **Referências relacionadas**

[Valores iniciais e declarações de idioma para MQWCR](#)

Valores iniciais e declarações de idioma C e High Level Assembler para MQWCR -Estrutura de registro de cluster de carga de trabalho do cluster.

## **Valores iniciais e declarações de idioma para MQWCR**

Valores iniciais e declarações de idioma C e High Level Assembler para MQWCR -Estrutura de registro de cluster de carga de trabalho do cluster.

*Tabela 834. Campos em MQWCR*

<b>Nome de Campo</b>	<b>Nome da constante</b>	<b>Valor da constante</b>
<i>ClusterName</i>	Nenhum	" "
<i>ClusterRecOffset</i>	Nenhum	0
<i>ClusterFlags</i>	Nenhum	0

## **Declaração C**

```
typedef struct tagMQWCR {
    MQCHAR48 ClusterName; /* Cluster name */
    MQLONG ClusterRecOffset; /* Offset of next cluster record */
    MQLONG ClusterFlags; /* Cluster flags */
};
```

## **High Level Assembler**

```
MQWCR          DSECT
MQWCR_CLUSTERNAME DS CL48 Cluster name
MQWCR_CLUSTERRECOFFSET DS F Offset of next cluster
* record
MQWCR_CLUSTERFLAGS DS F Cluster flags
MQWCR_LENGTH EQU *-MQWCR Length of structure
ORG MQWCR
MQWCR_AREA DS CL(MQWCR_LENGTH)
```

## **Referências relacionadas**

[Campos na estrutura de registro do cluster de carga de trabalho MQWCR -Cluster.](#)

[Descrição dos campos na estrutura de registro do cluster de carga de trabalho MQWCR -Cluster.](#)

## **Referência de saída de API**

Esta seção fornece informações de referência principalmente de interesse para um programador gravando saídas de API.

## Notas gerais de utilização

### notas:

1. Todas as funções de saída podem emitir a chamada MQXEP; esta chamada é projetada especificamente para uso a partir de funções de saída da API (interface de programação de aplicativos)
2. A função MQ\_INIT\_EXIT não pode emitir nenhuma chamada MQ diferente de MQXEP.
3. Não é possível emitir a chamada MQDISC para a conexão atual..
4. Se uma função de saída emitir a chamada MQCONN ou a chamada MQCONNX com a opção MQCNO\_HANDLE\_SHARE\_NONE, a chamada será concluída com o código de razão MQRC\_ALREADY\_CONNECTED e a manipulação retornada será a mesma que aquela transmitida para a saída como um parâmetro..
5. Em geral, quando uma função de saída de API emite uma chamada MQI, as saídas de API não são chamadas recursivamente.. No entanto, se uma função de saída emitir a chamada MQCONNX com as opções MQCNO\_HANDLE\_SHARE\_BLOCK ou MQCNO\_HANDLE\_SHARE\_NO\_BLOCK, a chamada retorna uma nova manipulação compartilhada. Isso fornece ao conjunto de saída um identificador de conexão próprio e, portanto, uma unidade de trabalho independente da unidade de trabalho do aplicativo. O conjunto de saída pode usar essa manipulação para colocar e obter mensagens dentro de sua própria unidade de trabalho e confirmar ou voltar para essa unidade de trabalho; tudo isso pode ser feito sem afetar a unidade de trabalho do aplicativo de forma alguma

Como a função de saída está usando um identificador de conexão diferente do identificador que está sendo usado pelo aplicativo, as chamadas do MQ emitidas pela função de saída resultam nas funções de saída da API relevantes sendo chamadas. As funções de saída podem, portanto, ser chamadas recursivamente. Observe que o campo *ExitUserArea* no MQAXP e a área da cadeia de saída têm escopo de manipulação de conexões. Consequentemente, uma função de saída não pode usar essas áreas para sinalizar para outra instância de si mesma chamada recursivamente que ela já está ativa.

6. As funções de saída também podem colocar e obter mensagens dentro da unidade de trabalho do aplicativo Quando o aplicativo confirma ou restaura a unidade de trabalho, todas as mensagens dentro da unidade de trabalho são confirmados ou restaurados juntos, independentemente de quem os colocou na unidade de trabalho (aplicativo ou função de saída). No entanto, a saída pode fazer com que o aplicativo exceda os limites do sistema mais cedo do que seria o caso (por exemplo, excedendo o número máximo de mensagens não confirmadas em uma unidade de trabalho).

Quando uma função de saída usa a unidade de trabalho do aplicativo dessa maneira, a função de saída geralmente deve evitar a emissão da chamada MQCMIT, pois isso confirma a unidade de trabalho do aplicativo e pode prejudicar o funcionamento correto do aplicativo. No entanto, a função de saída pode às vezes precisar emitir a chamada MQBACK, se a função de saída encontrar um erro grave que impeça a unidade de trabalho de ser confirmada (por exemplo, um erro colocando uma mensagem como parte da unidade de trabalho do aplicativo). Quando MQBACK for chamado, tome cuidado para assegurar que os limites da unidade de trabalho do aplicativo não sejam mudados Nesta situação, a função de saída deve configurar os valores apropriados para assegurar que o código de conclusão MQCC\_WARNING e o código de razão MQRC\_BACKED\_OUT sejam retornados ao aplicativo, para que o aplicativo possa detectar o fato de que a unidade de trabalho foi restaurada.

Se uma função de saída usar o identificador de conexão do aplicativo para emitir chamadas do MQ , essas próprias chamadas não resultarão em chamadas adicionais de funções de saída da API.

7. Se uma função de saída MQXR\_BEFORE for encerrada de forma anormal, o gerenciador de filas poderá ser capaz de se recuperar da falha. Se for possível, o gerenciador de filas continuará o processamento como se a função de saída tivesse retornado MQXCC\_FAILED. Se o gerenciador de filas não puder se recuperar, o aplicativo será finalizado
8. Se uma função de saída MQXR\_AFTER terminar de forma anormal, o gerenciador de filas poderá ser capaz de se recuperar da falha. Se for possível, o gerenciador de filas continuará o processamento como se a função de saída tivesse retornado MQXCC\_FAILED. Se o gerenciador de filas não puder se recuperar, o aplicativo será finalizado Esteja ciente de que, no último caso, as mensagens

recuperadas fora de uma unidade de trabalho são perdidas (essa é a mesma situação em que o aplicativo falhou imediatamente após a remoção de uma mensagem da fila).

9. O processo MCA executa um two-phase commit.

Se uma saída de API interceptar um MQCMIT de um processo MCA preparado e tentar executar uma ação dentro da unidade de trabalho, a ação falhará com o código de razão MQRC\_UOW\_NOT\_AVAILABLE.

10. Quando várias instalações do IBM MQ estiverem disponíveis, use as saídas gravadas para uma versão anterior do IBM MQ, já que a nova funcionalidade incluída na versão mais recente pode não funcionar com versões anteriores. Para obter mais informações sobre mudanças entre as liberações, consulte [O que mudou no IBM MQ 8.0](#).

## IBM MQ Estrutura do parâmetro de saída de API (MQAXP)

A estrutura MQAXP, um bloco de controle externo, é usado como um parâmetro de entrada ou de saída para a saída de API. Este tópico também fornece informações sobre como os gerenciadores de filas processam as funções de saídas.

MQAXP tem a seguinte declaração C:

```
typedef struct tagMQAXP {
    MQCHAR4   StrucId;           /* Structure identifier */
    MQLONG    Version;          /* Structure version number */
    MQLONG    ExitId;           /* Exit Identifier */
    MQLONG    ExitReason;       /* Exit invocation reason */
    MQLONG    ExitResponse;     /* Response code from exit */
    MQLONG    ExitResponse2;    /* Secondary response code from exit */
    MQLONG    Feedback;        /* Feedback code from exit */
    MQLONG    APICallerType;     /* MQSeries API caller type */
    MQBYTE16  ExitUserArea;     /* User area for use by exit */
    MQCHAR32  ExitData;         /* Exit data area */
    MQCHAR48  ExitInfoName;     /* Exit information name */
    MQBYTE48  ExitPDArea;      /* Problem determination area */
    MQCHAR48  QMgrName;        /* Name of local queue manager */
    PMQACH    ExitChainAreaPtr; /* Inter exit communication area */
    MQHCONFIG Hconfig;         /* Configuration handle */
    MQLONG    Function;        /* Function Identifier */
    /* Ver:1 */
    MQHMSG    ExitMsgHandle     /* Exit message handle */
    /* Ver:2 */
};
```

A lista de parâmetros a seguir é passada quando as funções em uma saída de API são chamadas:

### StrucId (MQCHAR4)-entrada

O identificador de estrutura do parâmetro de saída, com um valor de:

```
MQAXP_STRUC_ID.
```

O manipulador de saída configura esse campo na entrada para cada função de saída..

### Versão (MQLONG)-entrada

O número da versão da estrutura, com um valor de:

#### MQAXP\_VERSION\_1

Estrutura do parâmetro de saída da API Versão 1.

#### MQAXP\_VERSION\_2

Estrutura do parâmetro de saída da API Versão 2.

#### MQAXP\_CURRENT\_VERSION

O número da versão atual para a estrutura do parâmetro de saída da API (interface de programação de aplicativos)

O manipulador de saída configura esse campo na entrada para cada função de saída..

### ExitId (MQLONG)-entrada

O identificador de saída, definido na entrada para a rotina de saída, indicando o tipo de saída:

## **MQXT\_API\_EXIT**

Saída de API.

### **ExitReason (MQLONG)-entrada**

O motivo para chamar a saída, definido na entrada para cada função de saída:

#### **MQXR\_CONNECTION**

A saída está sendo chamada para inicializar-se antes de uma chamada MQCONN ou MQCONNX ou para terminar-se após uma chamada MQDISC.

#### **MQXR\_BEFORE**

A saída está sendo chamada antes de executar uma chamada API ou antes de converter dados em um MQGET.

#### **MQXR\_AFTER**

A saída está sendo chamada após a execução de uma chamada API.

### **ExitResponse (MQLONG)-saída**

A resposta da saída, inicializada na entrada para cada função de saída para:

#### **MQXCC\_OK**

Continue normalmente

Esse campo deve ser configurado pela função de saída, para comunicar ao gerenciador de filas o resultado da execução da função de saída O valor deve ser um dos seguintes:

#### **MQXCC\_OK**

A função de saída foi concluída com êxito Continue normalmente

Esse valor pode ser configurado por todas as funções de saída MQXR\_\*. ExitResponse2 é usado para decidir se as funções de saída devem ser chamadas posteriormente na cadeia.

#### **MQXCC\_FAILED**

A função de saída falhou por causa de um erro

Esse valor pode ser configurado por todas as funções de saída MQXR\_\*. O gerenciador de filas configura CompCode para MQCC\_FAILED e Reason para:

- MQRC\_API\_EXIT\_INIT\_ERROR se a função for MQ\_INIT\_EXIT
- MQRC\_API\_EXIT\_TERM\_ERROR se a função for MQ\_TERM\_EXIT
- MQRC\_API\_EXIT\_ERROR para todas as outras funções de saída

Os valores configurados podem ser alterados por uma função de saída posteriormente na cadeia.

ExitResponse2 é ignorado; o gerenciador de filas continua o processamento como se MQXR2\_SUPPRESS\_CHAIN tivesse sido retornado.

#### **MQXCC\_SUPPRESS\_FUNCTION**

Suprima a função da API IBM MQ

Esse valor pode ser configurado apenas por uma função de saída MQXR\_BEFORE. Ele ignora a chamada de API. Se for retornado pelo MQ\_DATA\_CONV\_ON\_GET\_EXIT, a conversão de dados será ignorada. O gerenciador de filas configura CompCode para MQCC\_FAILED e Reason para MQRC\_SUPPRESSED\_BY\_EXIT, mas os valores configurados podem ser alterados por uma função de saída posteriormente na cadeia.. Outros parâmetros para a chamada permanecem como a saída os deixou. ExitResponse2 é usado para decidir se as funções de saída devem ser chamadas posteriormente na cadeia.

Se esse valor for configurado por uma função de saída MQXR\_AFTER ou MQXR\_CONNECTION, o gerenciador de filas continuará o processamento como se MQXCC\_FAILED tivesse sido retornado..

#### **MQXCC\_SKIP\_FUNCTION**

Ignore a função da API IBM MQ .

Esse valor pode ser configurado apenas por uma função de saída MQXR\_BEFORE. Ele ignora a chamada de API. Se for retornado pelo MQ\_DATA\_CONV\_ON\_GET\_EXIT, a conversão de dados será ignorada. A função de saída deve configurar CompCode e Reason para os valores a serem

retornados ao aplicativo, mas o conjunto de valores pode ser alterado por uma função de saída posteriormente na cadeia. Outros parâmetros para a chamada permanecem como a saída os deixou. ExitResponse2 é usado para decidir se as funções de saída devem ser chamadas posteriormente na cadeia.

Se esse valor for configurado por uma função de saída MQXR\_AFTER ou MQXR\_CONNECTION, o gerenciador de filas continuará o processamento como se MQXCC\_FAILED tivesse sido retornado..

### **MQXCC\_SUPPRESS\_EXIT**

Suprimir todas as funções de saída pertencentes ao conjunto de saídas.

Esse valor pode ser configurado apenas pelas funções de saída MQXR\_BEFORE e MQXR\_AFTER. Ele ignora *todas* chamadas subsequentes de funções de saída pertencentes a esse conjunto de saídas para essa conexão lógica. Esse bypass continua até que a solicitação de desconexão lógica ocorra, quando a função MQ\_TERM\_EXIT é chamada com um ExitReason de MQXR\_CONNECTION.

A função de saída deve configurar CompCode e Reason para os valores a serem retornados ao aplicativo, mas o conjunto de valores pode ser alterado por uma função de saída posteriormente na cadeia. Outros parâmetros para a chamada permanecem como a saída os deixou. ExitResponse2 é ignorado.

Se esse valor for configurado por uma função de saída MQXR\_CONNECTION, o gerenciador de fila continuará processando como se MQXCC\_FAILED tivesse sido retornado..

Para obter informações sobre a interação entre ExitResponse e ExitResponse2e seu efeito no processamento de saída, consulte [“Como os gerenciadores de filas processam funções de saída” na página 1622.](#)

### **ExitResponse2 (MQLONG)-saída**

Este é um código de resposta de saída secundário que qualifica o código de saída primário para as funções de saída MQXR\_BEFORE. Ele é inicializado para:

```
MQXR2_DEFAULT_CONTINUATION
```

na entrada para uma função de saída de chamadas API do IBM MQ . Ele pode, então, ser configurado para um dos valores:

### **MQXR2\_DEFAULT\_CONTINUATION**

Se deve continuar com a próxima saída na cadeia, dependendo do valor de ExitResponse.

Se ExitResponse for MQXCC\_SUPPRESS\_FUNCTION ou MQXCC\_SKIP\_FUNCTION, ignore as funções de saída posteriormente na cadeia MQXR\_BEFORE e as funções de saída correspondentes na cadeia MQXR\_AFTER. Chame funções de saída na cadeia MQXR\_AFTER que correspondem às funções de saída anteriormente na cadeia MQXR\_BEFORE.

Caso contrário, chame a saída seguinte na cadeia.

### **MQXR2\_SUPPRESS\_CHAIN**

Suprimir a cadeia.

Ignore as funções de saída posteriormente na cadeia MQXR\_BEFORE e as funções de saída correspondentes na cadeia MQXR\_AFTER para essa chamada de API. Chame funções de saída na cadeia MQXR\_AFTER que correspondem às funções de saída anteriormente na cadeia MQXR\_BEFORE.

### **MQXR2\_CONTINUE\_CHAIN**

Continue com a próxima saída na cadeia.

Para obter informações sobre a interação entre ExitResponse e ExitResponse2e seu efeito no processamento de saída, consulte [“Como os gerenciadores de filas processam funções de saída” na página 1622.](#)

### **Feedback (MQLONG)-entrada/saída**

Comunicar códigos de feedback entre chamadas de função de saída. É inicializado para:



MQFB\_NONE (0)

antes de invocar a primeira função da primeira saída em uma cadeia.

As saídas podem configurar esse campo para qualquer valor, incluindo qualquer valor MQFB\_\* ou MQRC\_\* válido. As saídas também podem configurar esse campo para um valor de feedback definido pelo usuário no intervalo MQFB\_APPL\_FIRST para MQFB\_APPL\_LAST.

#### **APICallerType (MQLONG)-entrada**

O tipo de responsável pela chamada da API, indicando se o responsável pela chamada da API IBM MQ é externo ou interno para o gerenciador de filas: MQXACT\_EXTERNAL ou MQXACT\_INTERNAL.

#### **Área ExitUser(MQBYTE16)-entrada/saída**

Uma área do usuário, disponível para todas as saídas associadas a um Objeto ExitInfoespecífico... É inicializado para MQXUA\_NONE (zeros binários para o comprimento da Área ExitUser) antes de chamar a primeira função de saída (MQ\_INIT\_EXIT) para o hconn. A partir de então, todas as mudanças feitas nesse campo por uma função de saída são preservadas em chamadas de funções da mesma saída.

Esse campo está alinhado a um múltiplo de 4 MQLONGs

As saídas também podem ancorar qualquer armazenamento que elas alocam a partir desta área

Para cada hconn, cada saída em uma cadeia de saídas possui uma Área ExitUserdiferente. A área ExitUsernão pode ser compartilhada por saídas em uma cadeia e o conteúdo da área ExitUserpara uma saída não está disponível para outra saída em uma cadeia..

Para programas C, a constante MQXUA\_NONE\_ARRAY também é definida com o mesmo valor que MQXUA\_NONE, mas como uma matriz de caracteres, em vez de uma sequência.

O comprimento desse campo é fornecido por MQ\_EXIT\_USER\_AREA\_LENGTH

#### **ExitData (MQCHAR32)-entrada**

Dados de saída, configurados na entrada para cada função de saída para os 32 caracteres de dados específicos da saída fornecidos na saída. Se você não definir nenhum valor na saída, este campo estará todo em branco

O comprimento desse campo é fornecido por MQ\_EXIT\_DATA\_LENGTH

#### **ExitInfoNome (MQCHAR48)-entrada**

O nome das informações de saída, definido na entrada para cada função de saída para o ApiExit\_name especificado nas definições de saída nas sub-rotinas...

#### **ExitPDArea (MQBYTE48)-entrada / saída**

Uma área de determinação de problema, inicializado para MQXPDA\_NONE (zeros binários para o comprimento do campo) para cada chamada de uma função de saída..

Para programas C, a constante MQXPDA\_NONE\_ARRAY também é definida com o mesmo valor que MQXPDA\_NONE, mas como uma matriz de caracteres em vez de uma sequência...

O manipulador de saída sempre grava essa área no rastreamento IBM MQ no final de uma saída, mesmo quando a função é bem-sucedida.

O comprimento desse campo é fornecido por MQ\_EXIT\_PD\_AREA\_LENGTH

#### **QMgrName (MQCHAR48)-entrada de**

O nome do gerenciador de fila ao qual o aplicativo está conectado, que chamou uma saída como resultado do processamento de uma chamada da API IBM MQ

Se o nome de um gerenciador de filas fornecido em chamadas MQCONN ou MQCONNX estiver em branco, esse campo ainda será configurado para o nome do gerenciador de filas ao qual o aplicativo está conectado, se o aplicativo for servidor ou cliente.

O manipulador de saída configura esse campo na entrada para cada função de saída..

O comprimento desse campo é fornecido por MQ\_Q\_MGR\_NAME\_LENGTH

### **ExitChainAreaPtr (PMQACH)-entrada / saída**

Isso é usado para comunicar dados entre chamadas de diferentes saídas em uma cadeia. É configurado como um ponteiro NULL antes de chamar a primeira função (MQ\_INIT\_EXIT com ExitReason MQXR\_CONNECTION) da primeira saída em uma cadeia de saídas. O valor retornado pela saída em uma chamada é transmitido para a próxima chamada.

Consulte “A área da cadeia de saída e o cabeçalho da cadeia de saída (MQACH)” na página 1626 para obter mais detalhes sobre como usar a área da cadeia de saída

### **Hconfig (MQHCONFIG)-entrada**

O identificador de configuração, que representa o conjunto de funções que estão sendo inicializadas. Esse valor é gerado pelo gerenciador de filas na função MQ\_INIT\_EXIT e é transmitido posteriormente para a função de saída de API. É configurado na entrada para cada função de saída.

É possível usar Hconfig como ponteiro para a estrutura MQIEP para fazer chamadas MQI e DCI. Você deve verificar se os primeiros 4 bytes de HConfig correspondem ao StrucId da estrutura MQIEP antes de usar o parâmetro HConfig como um indicador para a estrutura MQIEP.

### **Função (MQLONG)-entrada**

O identificador de função, cujos valores válidos são as constantes MQXF\_\* descritas em “Constantes externas” na página 1627.

O manipulador de saída configura esse campo para o valor correto, na entrada para cada função de saída, dependendo da chamada API IBM MQ que resultou na saída sendo chamada.

### **ExitMsgHandle (MQHMSG)-entrada/saída**

Quando a Função é MQXF\_GET e ExitReason é MQXR\_AFTER, uma manipulação de mensagens válida é retornada nesse campo, permitindo que a saída de API acesse os campos do descritor de mensagens e quaisquer outras propriedades correspondentes à sequência ExitProperties especificada na estrutura MQXEPO ao registrar a saída de API.

Quaisquer propriedades do descritor de não mensagens que são retornadas na Manipulação ExitMsg não estarão disponíveis a partir de MsgHandle na estrutura MQGMO se uma tiver sido especificada ou nos dados da mensagem.

Quando a Função for MQXF\_GET e ExitReason for MQXR\_BEFORE, se o programa de saída configurar esse campo como MQHM\_NONE, ele suprimirá o preenchimento das propriedades da Identificação ExitMsg.

Esse campo não será configurado se a Versão for menor que MQAXP\_VERSION\_2.

## **Como os gerenciadores de filas processam funções de saída**

O processamento executado pelo gerenciador de filas no retorno de uma função de saída depende de ExitResponse e ExitResponse2.

Tabela 835 na página 1623 resume as combinações possíveis e seus efeitos para uma função de saída MQXR\_BEFORE, mostrando:

- Quem configura os parâmetros CompCode e Reason da chamada API
- Se as funções de saída restantes na cadeia MQXR\_BEFORE e as funções de saída correspondentes na cadeia MQXR\_AFTER são chamadas
- Se a chamada da API é chamada

Para uma função de saída MQXR\_AFTER:

- CompCode e Reason são configurados da mesma maneira que MQXR\_BEFORE
- ExitResponse2 é ignorado (as funções de saída restantes na cadeia MQXR\_AFTER são sempre chamadas)
- MQXCC\_SUPPRESS\_FUNCTION e MQXCC\_SKIP\_FUNCTION não são válidos

Para uma função de saída MQXR\_CONNECTION:

- CompCode e Reason são configurados da mesma maneira que MQXR\_BEFORE

- ExitResponse2 é ignorado
- MQXCC\_SUPPRESS\_FUNCTION, MQXCC\_SKIP\_FUNCTION, MQXCC\_SUPPRESS\_EXIT não são válidos

Em todos os casos, em que uma saída ou o gerenciador de filas configura CompCode e Reason, o conjunto de valores pode ser alterado por uma saída chamada posteriormente ou pela chamada API (se a chamada API for chamada posteriormente).

*Tabela 835. Processamento de saída MQXR\_BEFORE*

Valor de ExitResponse	CompCode e Motivo configurado por	Valor de ExitResponse2 (continuação padrão) Cadeia	Valor da API ExitResponse2 (continuação padrão)
MQXCC_OK	saída	Y	Y
MQXCC_SUPPRESS_EXIT	saída	Y	Y
MQXCC_SUPPRESS_FUNCTION	gerenciador de filas	N	N
FUNÇÃO MQXCC_SKIP	saída	N	N
MQXCC_FAILED	gerenciador de filas	N	N

## Como os clientes processam as funções de saída

Em geral, os clientes processam as funções de saída da mesma maneira que os aplicativos do servidor e o atributo *QMGrName* nessa estrutura se aplica se a função estiver em um servidor ou em um cliente.

No entanto, o cliente não tem conceito do arquivo *mqs.ini*, portanto as sub-rotinas *ApiExitCommon* e *APIExitTemplate* não se aplicam. Apenas a sub-rotina *ApiExitLocal* se aplica e essa sub-rotina é configurada no arquivo *mclient.ini*

## IBM MQ estrutura de contexto de saída da API (MQAXC)

A estrutura MQAXC, um bloco de controle externo, é usado como um parâmetro de entrada para uma saída de API.

MQAXC possui a seguinte declaração C:

```
typedef struct tagMQAXC {
    MQCHAR4   StrucId;           /* Structure identifier */
    MQLONG    Version;          /* Structure version number */
    MQLONG    Environment;      /* Environment */
    MQCHAR12  UserId;           /* UserId associated with appl */
    MQBYTE40  SecurityId        /* Extension to UserId running appl */
    MQCHAR264 ConnectionName;    /* Connection name */
    MQLONG    LongMCAUserIdLength; /* long MCA user identifier length */
    MQLONG    LongRemoteUserIdLength; /* long remote user identifier length */
    MQPTR     LongMCAUserIdPtr;  /* long MCA user identifier address */
    MQPTR     LongRemoteUserIdPtr; /* long remote user identifier address */
    MQCHAR28  ApplName;         /* Application name */
    MQLONG    ApplType;         /* Application type */
    MQPID     ProcessId;        /* Process identifier */
    MQTID     ThreadId;         /* Thread identifier */

    /* Ver:1 */
    MQCHAR    ChannelName[20]   /* Channel Name */
    MQBYTE4   Reserved1;        /* Reserved */
    PMQCD     pChannelDefinition; /* Channel Definition pointer */
};
```

Os parâmetros para MQAXC são:

**StrucId (MQCHAR4)-entrada**

O identificador da estrutura de contexto de entrada, com um valor de MQAXC\_STRUC\_ID. Para programas C, a constante MQAXC\_STRUC\_ID\_ARRAY também é definida, com o mesmo valor que MQAXC\_STRUC\_ID, mas como uma matriz de caracteres, em vez de uma sequência.

O manipulador de saída configura esse campo na entrada para cada função de saída..

**Versão (MQLONG)-entrada**

O número da versão da estrutura, com um valor de:

**MQAXC\_VERSION\_2**

Número da versão para a estrutura de contexto de saída..

**MQAXC\_CURRENT\_VERSION**

Número da versão atual da estrutura do contexto de saída.

O manipulador de saída configura esse campo na entrada para cada função de saída..

**Ambiente (MQLONG)-entrada de**

O ambiente a partir do qual uma chamada API IBM MQ foi emitida que resultou em uma função de saída sendo acionada. Os valores válidos para esse campo são:

**MQXE\_OTHER**

Este valor é consistente com as chamadas que uma saída de API vê se a saída é chamada de um aplicativo do servidor. Isso significa que uma saída API é executada inalterada em um cliente e não vê nada diferente.

Se a saída realmente precisar determinar se está em execução no cliente, a saída poderá fazer isso consultando os campos *ChannelName* e *ChannelDefinition* .

**MQXE\_MCA**

Agente do canal de mensagens

**MQXE\_MCA\_SVRCONN**

Um agente do canal de mensagens agindo em nome de um cliente

**MQXE\_COMMAND\_SERVER**

O servidor de comandos

**MQXE\_MQSC**

O interpretador do comando runmqsc

O manipulador de saída configura esse campo na entrada para cada função de saída..

**UserId (MQCHAR12)-entrada**

O ID do usuário associado ao aplicativo. Em particular, no caso de conexões do cliente, esse campo contém o ID do usuário do usuário adotado em oposição ao ID do usuário sob o qual o código do canal está em execução. Se um ID do usuário em branco flui do cliente, nenhuma mudança será feita para o ID do usuário já sendo usado. Ou seja, nenhum novo ID do usuário é adotado

O manipulador de saída configura esse campo na entrada para cada função de saída.. O comprimento desse campo é fornecido por MQ\_USER\_ID\_LENGTH

No caso de um cliente, este é o ID do usuário enviado do cliente ao servidor. Observe que este pode não ser o ID do usuário efetivo com relação ao qual o cliente está em execução no gerenciador de filas, pois poderia haver uma configuração MCAUser ou CHLAUTH que altera o ID do usuário.

**SecurityId (MQBYTE40)-entrada**

Uma extensão para o ID do usuário executando o aplicativo. Seu comprimento é fornecido por MQ\_SECURITY\_ID\_LENGTH

No caso de um cliente, este é o ID do usuário enviado do cliente ao servidor. Observe que este pode não ser o ID do usuário efetivo com relação ao qual o cliente está em execução no gerenciador de filas, pois poderia haver uma configuração MCAUser ou CHLAUTH que altera o ID do usuário.

**ConnectionName (MQCHAR264)-entrada de**

O campo de nome de conexão, configurado para o endereço do cliente Por exemplo, para TCP/IP, seria o endereço IP do cliente.

O comprimento desse campo é fornecido por MQ\_CONN\_NAME\_LENGTH

No caso de um cliente, este é o endereço do parceiro do gerenciador de fila.

#### **LongMCAUserIdLength (MQLONG)-entrada**

O comprimento do identificador de usuários MCA longo.

Quando o MCA se conecta ao gerenciador de fila, esse campo é configurado para o comprimento do identificador de usuário do MCA longo (ou zero se não houver esse identificador)...

No caso de um cliente, este é o identificador de usuário longo do cliente..

#### **LongRemoteUserIdComprimento (MQLONG)-entrada**

O comprimento do identificador de usuário remoto longo..

Quando o MCA se conecta ao gerenciador de filas, esse campo é configurado para o comprimento do identificador de usuário remoto longo.. Caso contrário, esse campo será configurado como zero.

No caso de um cliente, configure esse campo como zero.

#### **LongMCAUserIdPtr (MQPTR)-entrada**

Endereço do identificador de usuário MCA longo.

Quando o MCA se conecta ao gerenciador de filas, esse campo é configurado para o endereço do identificador de usuário longo do MCA (ou para um ponteiro nulo se não houver esse identificador).

No caso de um cliente, este é o identificador de usuário longo do cliente..

#### **LongRemoteUserIdPtr (MQPTR)-entrada**

O endereço do identificador de usuário remoto longo.

Quando o MCA se conecta ao gerenciador de filas, esse campo é configurado para o endereço do identificador de usuário remoto longo (ou para um ponteiro nulo se não houver esse identificador).

No caso de um cliente, configure esse campo como zero.

#### **ApplName (MQCHAR28)-entrada**

O nome do aplicativo ou componente que emitiu a chamada API IBM MQ .

As regras para gerar o ApplName são as mesmas que para gerar o nome padrão para um MQPUT.

O valor deste campo é encontrado consultando o nome do programa no sistema operacional. Seu comprimento é fornecido por MQ\_APPL\_NAME\_LENGTH

#### **ApplType (MQLONG)-entrada de**

O tipo de aplicativo ou componente que emitiu a chamada da API IBM MQ

O valor é MQAT\_DEFAULT para a plataforma na qual o aplicativo é compilado ou é igual a um dos valores MQAT\_\* definidos.

O manipulador de saída configura esse campo na entrada para cada função de saída..

#### **ProcessId (MQPID)-entrada**

O identificador do processo do sistema operacional.

Quando aplicável, o manipulador de saída configura este campo na entrada para cada função de saída..

#### **ThreadId (MQTID)-entrada**

O identificador de encadeamento do MQ . Esse é o mesmo identificador usado no rastreamento do MQ e FFST dumps, mas pode ser diferente do identificador de encadeamento do sistema operacional.

Quando aplicável, o manipulador de saída configura este campo na entrada para cada função de saída..

#### **ChannelName (MQCHAR)-entrada**

O nome do canal, preenchido com espaços em branco, se aplicável e conhecido.

Se não for aplicável, esse campo será configurado como caracteres NULL.

### Reserved1 (MQBYTE4)-entrada

Este campo é reservado

### ChanneDefinition (PMQCD)-entrada

Um ponteiro para a definição de canal sendo usada, se aplicável e conhecido.

Se não for aplicável, esse campo será configurado como caracteres NULL.

Observe que o ponteiro só será concluído se a conexão estiver sendo processada em nome de um canal do IBM MQ e essa definição de canal tiver sido lida.

Em particular, a definição de canal não é dada no servidor quando a primeira chamada MQCONN é feita para o canal.. Além disso, se o ponteiro for preenchido, a estrutura (e quaisquer subestruturas) apontada pelo ponteiro deve ser tratada como somente leitura; qualquer atualização da estrutura levaria a resultados imprevisíveis e não é suportada.

No caso de um cliente, campos diferentes daqueles com um valor especificado para um cliente, contêm valores que são apropriados para um aplicativo cliente.

## A área da cadeia de saída e o cabeçalho da cadeia de saída (MQACH)

Se necessário, uma função de saída pode adquirir armazenamento para uma área da cadeia de saída e configurar ExitChainAreaPtr em MQAXP para apontar para esse armazenamento.

Saídas (as mesmas ou diferentes funções de saída) podem adquirir várias áreas da cadeia de saída e vinculá-las juntas. As áreas da cadeia de saída só devem ser incluídas ou removidas desta lista enquanto forem chamadas do manipulador de saída. Isso assegura que não haja problemas de serialização causados por diferentes encadeamentos incluindo ou removendo áreas da lista ao mesmo tempo..

Uma área de cadeia de saída deve iniciar com uma estrutura de cabeçalho MQACH, cuja declaração C é:

```
typedef struct tagMQACH {
    MQCHAR4   StrucId;           /* Structure identifier */
    MQLONG    Version;          /* Structure version number */
    MQLONG    StrucLength;      /* Length of the MQACH structure */
    MQLONG    ChainAreaLength;  /* Exit chain area length */
    MQCHAR48  ExitInfoName     /* Exit information name */
    PMQACH    NextChainAreaPtr; /* Pointer to next exit chain area */
};
```

Os campos no cabeçalho da área da cadeia de saída são:

### StrucId (MQCHAR4)-entrada

O identificador da estrutura de área da sequência de saída, com um valor inicial, definido por MQACH\_DEFAULT, de MQACH\_STRUC\_ID.

Para programas C, a constante MQACH\_STRUC\_ID\_ARRAY também é definida; isso tem o mesmo valor que MQACH\_STRUC\_ID, mas como uma matriz de caracteres em vez de uma sequência.

### Versão (MQLONG)-entrada

O número da versão da estrutura, como a seguir:

#### MQACH\_VERSION\_1

O número da versão para a estrutura do parâmetro de saída

#### MQACH\_CURRENT\_VERSION

O número da versão atual da estrutura do contexto de saída.

O valor inicial desse campo, definido por MQACH\_DEFAULT, é MQACH\_CURRENT\_VERSION.

**Nota:** Se você introduzir uma nova versão dessa estrutura, o layout da parte existente não mudará. As funções de saída devem verificar se o número da versão é igual ou maior que a versão mais baixa contendo os campos que a função de saída precisa usar.

### StrucLength (MQLONG)-entrada

O comprimento da estrutura MQACH.. As saídas podem usar esse campo para determinar o início dos dados de saída, definindo-o para o comprimento da estrutura criada pela saída

O valor inicial desse campo, definido por MQACH\_DEFAULT, é MQACH\_CURRENT\_LENGTH.

### Comprimento de ChainArea(MQLONG)-entrada

O comprimento da área da cadeia de saída, configurado para o comprimento geral da área da cadeia de saída atual, incluindo o cabeçalho MQACH

O valor inicial desse campo, definido por MQACH\_DEFAULT, é zero

### ExitInfoNome (MQCHAR48)-entrada

O nome das informações de saída

Quando uma saída cria uma estrutura MQACH, ela deve inicializar esse campo com seu próprio Nome ExitInfo, para que posteriormente essa estrutura MQACH possa ser localizada por outra instância dessa saída ou por uma saída de cooperação.

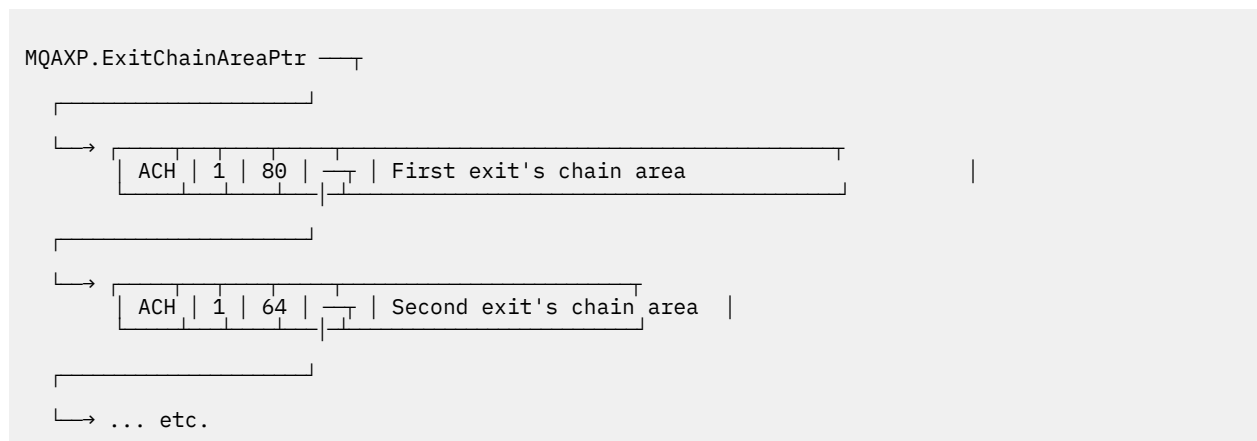
O valor inicial desse campo, definido por MQACH\_DEFAULT, é uma sequência de comprimento zero ({}).

### NextChainAreaPtr (PMQACH)-entrada

Um ponteiro para a próxima área da cadeia de saída com um valor inicial, definido por MQACH\_DEFAULT, de ponteiro nulo (NULL)..

As funções de saída devem liberar o armazenamento para quaisquer áreas de cadeia de saída que elas adquirirem e manipular os ponteiros de cadeia para remover suas áreas de cadeia de saída da lista..

Uma área da cadeia de saída pode ser construída da seguinte forma:



## Constantes externas

Use este tópico como informações de referência para constantes externas disponíveis para a API.

As constantes externas a seguir estão disponíveis para saídas de API:

### MQXF\_\* (identificadores de função de saída).

MQXF_INIT	1	X'00000001'
MQXF_TERM	2	X'00000002'
MQXF_CONN	3	X'00000003'
MQXF_CONNX	4	X'00000004'
MQXF_DISC	5	X'00000005'
MQXF_OPEN	6	X'00000006'
MQXF_CLOSE	7	X'00000007'
MQXF_PUT1	8	X'00000008'
MQXF_PUT	9	X'00000009'
MQXF_GET	10	X'0000000A'
MQXF_DATA_CONV_ON_GET	11	X'0000000B'
MQXF_INQ	12	X'0000000C'
MQXF_SET	13	X'0000000D'
MQXF_BEGIN	14	X'0000000E'
MQXF_CMIT	15	X'0000000F'
MQXF_BACK	16	X'00000010'
MQXF_STAT	18	X'00000012'
MQXF_CB	19	X'00000013'
MQXF_CTL	20	X'00000014'

MQXF_CALLBACK	21	X'00000015'
MQXF_SUB	22	X'00000016'
MQXF_SUBRQ	23	X'00000017'
MQXF_XACLOSE	24	X'00000018'
MQXF_XACOMMIT	25	X'00000019'
MQXF_XACOMPLETE	26	X'0000001A'
MQXF_XAEND	27	X'0000001B'
MQXF_XAFORGET	28	X'0000001C'
MQXF_XAOPEN	29	X'0000001D'
MQXF_XAPREPARE	30	X'0000001E'
MQXF_XARECOVER	31	X'0000001F'
MQXF_XAROLLBACK	32	X'00000020'
MQXF_XASTART	33	X'00000021'
MQXF_AXREG	34	X'00000022'
MQXF_AXUNREG	35	X'00000023'

### MQXR\_\* (motivos de saída).

MQXR_BEFORE	1	X'00000001'
MQXR_AFTER	2	X'00000002'
MQXR_CONNECTION	3	X'00000003'

### MQXE\_\* (ambientes)

MQXE_OTHER	0	X'00000000'
MQXE_MCA	1	X'00000001'
MQXE_MCA_SVRCONN	2	X'00000002'
MQXE_COMMAND_SERVER	3	X'00000003'
MQXE_MQSC	4	X'00000004'

### MQ\*\_\* (constantes adicionais)

MQAXP_VERSION_1	1	
MQAXP_VERSION_2	2	
MQAXC_VERSION_1	1	
MQACH_VERSION_1	1	
MQAXP_CURRENT_VERSION	1	
MQAXC_CURRENT_VERSION	1	
MQACH_CURRENT_VERSION	1	
MQXACT_EXTERNAL	1	
MQXACT_INTERNAL	2	
MQXT_API_EXIT	2	
MQACH_LENGTH_1	68 (32-bit platforms) 72 (64-bit platforms) 80 (128-bit platforms)	
MQACH_CURRENT_LENGTH	68 (32-bit platforms) 72 (64-bit platforms) 80 (128-bit platforms)	

### MQ\*\_\* (constantes nulas)

MQXPDA_NONE	X'00...00' (48 nulls)
MQXPDA_NONE_ARRAY	'\0','\0',...,'\0','\0'

### MQXCC\_\* (códigos de conclusão)

MQXCC_FAILED	-8
--------------	----

### MQRC\_\* (códigos de razão)

#### MQRC\_API\_EXIT\_ERROR 2374 X'00000946'

Uma chamada de função de saída retornou um código de resposta inválido ou falhou de alguma maneira, e o gerenciador de filas não pode determinar a próxima ação a ser executada..



Examine os campos ExitResponse e ExitResponse2 do MQAXP para determinar o código de resposta inválido e altere a saída para retornar um código de resposta válido.

**MQRC\_API\_EXIT\_INIT\_ERROR 2375 X'00000947'**

O gerenciador de filas encontrou um erro ao inicializar o ambiente de execução para uma função de saída de API.

**MQRC\_API\_EXIT\_TERM\_ERROR 2376 X'00000948'**

O gerenciador de filas encontrou um erro durante o fechamento do ambiente de execução para uma função de saída de API.

**MQRC\_EXIT\_REASON\_ERROR 2377 X'00000949'**

O valor do campo ExitReason fornecido em uma chamada de registro de ponto de entrada de saída (MQXEP) está com erro.

Examine o valor do campo ExitReason para determinar e corrigir o valor de razão de saída inválido...

**MQRC\_RESERVED\_VALUE\_ERROR 2378 X'0000094A'**

O valor do campo Reservado está com erro

Examine o valor do campo Reservado para determinar e corrigir o valor Reservado.

## C linguagem typedefs

Este tópico fornece informações sobre typedefs associados a saídas de API disponíveis na linguagem C.

Aqui estão os typedefs de linguagem C associados às saídas de API:

```
typedef PMQLONG MQPOINTER PPMQLONG;
typedef PMQBYTE MQPOINTER PPMQBYTE;
typedef PMQHOBJS MQPOINTER PPMQHOBJS;
typedef PMQOD MQPOINTER PPMQOD;
typedef PMQMD MQPOINTER PPMQMD;
typedef PMQPMO MQPOINTER PPMQPMO;
typedef PMQGMO MQPOINTER PPMQGMO;
typedef PMQCNO MQPOINTER PPMQCNO;
typedef PMQBO MQPOINTER PPMQBO;

typedef MQAXP MQPOINTER PMQAXP;
typedef MQACH MQPOINTER PMQACH;
typedef MQAXC MQPOINTER PMQAXC;

typedef MQCHAR MQCHAR16[16];
typedef MQCHAR16 MQPOINTER PMQCHAR16;

typedef MQLONG MQPID;
typedef MQLONG MQTID;
```

## A chamada de registro do ponto de entrada de saída (MQXEP)

Use estas informações para aprender sobre MQXEP, chamada de linguagem MQXEP C e protótipo de função MQXEP C.

Use a chamada MQXEP para:

1. Registre os pontos de chamada de saída da API IBM MQ anteriores e posteriores nos quais chamar funções de saída
2. Especifique os pontos de entrada da função de saída
3. Remover registro dos pontos de entrada da função de saída

Você normalmente codificaria as chamadas MQXEP na função de saída MQ\_INIT\_EXIT, mas é possível especificá-las em qualquer função de saída subsequente.

Se você usar uma chamada MQXEP para registrar uma função de saída já registrada, a segunda chamada MQXEP será concluída com êxito, substituindo a função de saída registrada.

Se você usar uma chamada MQXEP para registrar uma função de saída NULL, a chamada MQXEP será concluída com êxito e a função de saída terá o registro removido.

Se chamadas MQXEP forem usadas para registrar, remover o registro e registrar novamente uma função de saída específica durante a vida de uma solicitação de conexão, a função de saída registrada anteriormente será reativada. Qualquer armazenamento ainda alocado e associado a esta instância de função de saída está disponível para uso pelas funções da saída... (Esse armazenamento geralmente é liberado durante a chamada da função de saída de término)

A interface para MQXEP é:

```
MQXEP (Hconfig, ExitReason, Function, EntryPoint, &ExitOpts, &CompCode, &Reason)
```

em que:

#### **Hconfig (MQHCONFIG)-entrada**

O identificador de configuração, que representa a saída de API que inclui o conjunto de funções que está sendo inicializado. Esse valor é gerado pelo gerenciador de filas imediatamente antes de chamar a função MQ\_INIT\_EXIT e é transmitido no MQAXP para cada função de saída de API...

#### **ExitReason (MQLONG)-entrada**

O motivo pelo qual o ponto de entrada está sendo registrado, a partir dos seguintes motivos:

- Inicialização ou finalização do nível de conexão (MQXR\_CONNECTION).
- Antes de uma chamada API IBM MQ (MQXR\_BEFORE)
- Após uma chamada API IBM MQ (MQXR\_AFTER)

#### **Função (MQLONG)-entrada**

O identificador de função, valores válidos para os quais são as constantes MQXF\_\* (consulte [“Constantes externas” na página 1627](#)).

#### **EntryPoint (PMQFUNC)-entrada**

O endereço do ponto de entrada para a função de saída a ser registrada. O valor NULL indica que a função de saída não foi fornecida ou que um registro anterior da função de saída está tendo o registro removido.

#### **ExitOpts(MQXEPO)**

Saídas de API podem especificar opções que controlam como as saídas de API são registradas. Se um ponteiro nulo for especificado para esse campo, os valores padrão da estrutura MQXEPO serão assumidos..

#### **CompCode (MQLONG)-saída**

O código de conclusão, os valores válidos para os quais são:

##### **MQCC\_OK**

Indica conclusão bem-sucedida.

##### **MQCC\_FAILED**

A chamada falhou.

#### **Motivo (MQLONG)-saída**

O código de razão que qualifica o código de finalização

Se o código de conclusão for MQCC\_OK:

##### **MQRC\_NONE**

(0, X'000') Nenhuma razão a relatar.

Se o código de conclusão for MQCC\_FAILED

##### **MQRC\_HCONFIG\_ERROR**

(2280, X'8E8') A manipulação de configuração fornecida não é válida. Use a manipulação de configuração do MQAXP.

##### **MQRC\_EXIT\_REASON\_ERROR**

(2377, X'949') A razão de chamada da função de saída fornecida não é válida ou não é válida para o identificador da função de saída fornecido..

Use uma das razões de chamada de função de saída válidas (valor MQXR\_\*) ou use uma combinação válida de identificador de função e de razão de saída (Consulte Tabela 836 na página 1631.)

### **MQRC\_FUNCTION\_ERROR**

(2281, X'8E9') O identificador de função fornecido não é válido para a razão da saída da API. A tabela a seguir mostra combinações válidas de identificadores de função e ExitReasons.

<i>Tabela 836. Combinações válidas de identificadores de função e ExitReasons</i>	
<b>Função</b>	<b>ExitReason</b>
MQXF_INIT MQXF_TERM	MQXR_CONNECTION
MQXF_CONN MQXF_CONNX MQXF_DISC MQXF_OPEN MQXF_CLOSE MQXF_PUT1 MQXF_PUT MQXF_GET MQXF_INQ MQXF_SET MQXF_BEGIN MQXF_CMIT MQXF_BACK MQXF_STAT MQXF_CB MQXF_CTL MQXF_CALLBACK MQXF_SUB MQXF_SUBRQ	MQXR_BEFORE MQXR_AFTER
MQXF_DATA_CONV_ON_GET	MQXR_BEFORE

### **MQRC\_RESOURCE\_PROBLEM**

(2102, X'836 ') Uma tentativa de registrar ou cancelar registro de uma função de saída falhou devido a um problema de recurso.

### **MQRC\_UNEXPECTED\_ERROR**

(2195, X'893 ') Uma tentativa de registrar ou remover o registro de uma função de saída falhou inesperadamente..

### **MQRC\_PROPERTY\_NAME\_ERROR**

(2442, X'098A') Nome ExitProperties inválido.

### **MQRC\_XEPO\_ERROR**

(2507, X'09CB') Estrutura de opções de saída inválida.

## **Chamada de linguagem MQXEP C.**

```
MQXEP (Hconfig, ExitReason, Function, EntryPoint, &ExitOpts, &CompCode, &Reason);
```

Declaração para lista de parâmetros:

```
MQHCONFIG      Hconfig;          /* Configuration handle */
MQLONG         ExitReason;     /* Exit reason */
MQLONG         Function;       /* Function identifier */
```

```

PMQFUNC      EntryPoint;      /* Function entry point */
MQXEPO       ExitOpts;       /* Options that control the action of MQXEP */
MQLONG       CompCode;       /* Completion code */
MQLONG       Reason;        /* Reason code qualifying completion
                             code */

```

## Protótipo da função MQXEP C

```

void MQXEP (
MQHCONFIG    Hconfig,        /* Configuration handle */
MQLONG       ExitReason,     /* Exit reason */
MQLONG       Function,      /* Function identifier */
PMQFUNC      EntryPoint,     /* Function entry point */
MQXEPO       pExitOpts;     /* Options that control the action of MQXEP */
PMQLONG      pCompCode,     /* Address of completion code */
PMQLONG      pReason);     /* Address of reason code qualifying completion
                             code */

```

## Funções de saída..

Esta seção fornece algumas informações gerais para ajudá-lo ao usar chamadas de função e descreve como chamar as funções de saída individuais.

Use estas informações para entender as regras gerais para as rotinas de saída de API e para configurar e limpar o ambiente de execução de saída de saída.

## Regras gerais para rotinas de saída de API

As regras gerais a seguir se aplicam ao chamar rotinas de saída de API:

- Em todos os casos, as funções de saída da API são orientadas antes de validar os parâmetros de chamada da API e antes de quaisquer verificações de segurança (no caso de MQCONN, MQCONNX ou MQOPEN).
- Os valores de campos inseridos e de saída de uma rotina de saída são:
  - Na entrada para uma função de saída da API *antes* IBM MQ , o valor de um campo pode ser configurado pelo programa aplicativo ou por uma chamada de função de saída anterior.
  - Na saída de uma função de saída da API *antes* IBM MQ , o valor de um campo pode ser deixado inalterado ou configurado como algum outro valor pela função de saída..
  - Na entrada para uma função de saída de API *após* IBM MQ , o valor de um campo pode ser o valor configurado pelo gerenciador de filas após o processamento da chamada API IBM MQ ou pode ser configurado para um valor por uma chamada de função de saída anterior na cadeia de funções de saída.
  - Na saída de uma função de saída de chamada API *after* IBM MQ , o valor de um campo pode ser deixado inalterado ou configurado para algum outro valor pela função de saída.
- As funções de saída devem se comunicar com o gerenciador de filas usando os campos ExitResponse e ExitResponse2 .
- Os campos CompCode e Código de Razão se comunicam de volta para o aplicativo As funções do gerenciador e de saída podem configurar os campos CompCode e Código de Razão.
- A chamada MQXEP retorna novos códigos de razão às funções de saída que chamam MQXEP. No entanto, as funções de saída podem converter esses novos códigos de razão em quaisquer códigos de razão existentes que os aplicativos novos e existentes possam entender
- Cada protótipo de função de saída possui parâmetros semelhantes à função de API com um nível extra de indireção, exceto para o CompCode e o Reason.
- As saídas de API podem emitir chamadas MQI (exceto MQDISC), mas essas chamadas MQI não chamam saídas de API.

Observe que se o aplicativo estiver em um servidor ou em um cliente, não será possível prever o sequenciamento das chamadas de saída de API. Uma chamada BEFORE de saída de API pode não ser seguida imediatamente por uma chamada AFTER

A chamada BEFORE pode ser seguida por outra chamada BEFORE . Por exemplo:

```
ANTES de MQCTL
Retorno de Chamada BEFORE
ANTES DO MQPUT
AFTER MQPUT
Retorno de Chamada AFTER
AFTER MQCTL
```

ou

```
ANTES DE XAOPEN
ANTES de MQCONNX
AFTER MQCONNX
APÓS XAOPEN
```

No cliente, existe uma saída que pode modificar o comportamento da chamada MQCONN ou MQCONNX, chamada de saída PreConnect . A saída PreConnect pode modificar qualquer um dos parâmetros na chamada MQCONN ou MQCONNX incluindo o nome do gerenciador de filas. O cliente chama essa saída primeiro e, em seguida, chama a chamada MQCONN ou MQCONNX Observe que apenas a chamada inicial MQCONN ou MQCONNX chama a saída de API; quaisquer chamadas de reconexão subsequentes não têm efeito..

## O ambiente de execução

Em geral, todos os erros das funções de saída são comunicados de volta ao manipulador de saída usando os campos ExitResponse e ExitResponse2 em MQAXP.

Esses erros, por sua vez, são convertidos em valores MQCC\_\* e MQRC\_\* e comunicados de volta para o aplicativo nos campos CompCode e Razão No entanto, quaisquer erros encontrados na lógica do manipulador de saída são comunicados de volta ao aplicativo como valores MQCC\_\* e MQRC\_\* nos campos CompCode e Reason.

Se uma função MQ\_TERM\_EXIT retornar um erro:

- A chamada MQDISC já ocorreu
- Não há outra oportunidade para conduzir a *após* a função de saída MQ\_TERM\_EXIT (e, assim, executar a limpeza do ambiente de execução de saída)
- A limpeza do ambiente de execução de saída não é executada..

A saída não pode ser descarregada, pois ainda pode estar em uso Além disso, outras saídas registradas mais abaixo na cadeia de saída para as quais a saída *antes* foi bem-sucedida, serão conduzidas na ordem reversa

## Configurando o Ambiente de Execução de Saída

Ao processar uma chamada MQCONN ou MQCONNX explícita, a lógica de manipulação de saída configura o ambiente de execução de saída antes de chamar a função de inicialização de saída (MQ\_INIT\_EXIT). A configuração do ambiente de execução de saída envolve carregar a saída, adquirir armazenamento para e inicializar estruturas de parâmetros de saída. O identificador de configuração de saída também é alocado

Se ocorrerem erros durante essa fase, a chamada MQCONN ou MQCONNX falhará com CompCode MQCC\_FAILED e um dos seguintes códigos de razão:

### **MQRC\_API\_EXIT\_LOAD\_ERROR**

Uma tentativa de carregar um módulo de saída API falhou.

**MQRC\_API\_EXIT\_NOT\_FOUND**

Uma função de saída API não pôde ser localizada no módulo de saída API.

**MQRC\_STORAGE\_NOT\_AVAILABLE**

Uma tentativa de inicializar o ambiente de execução para uma função de saída de API falhou porque havia armazenamento insuficiente disponível.

**MQRC\_API\_EXIT\_INIT\_ERROR**

Foi encontrado um erro ao inicializar o ambiente de execução para uma função de saída de API..

**Limpendo o ambiente de execução de saída**

Ao processar uma chamada MQDISC explícita ou uma solicitação de desconexão implícita como resultado de um término de aplicativo, a lógica de manipulação de saída pode precisar limpar o ambiente de execução de saída depois de chamar a função de término de saída (MQ\_TERM\_EXIT), se registrada.

A limpeza do ambiente de execução de saída envolve liberar armazenamento para estruturas de parâmetros de saída, possivelmente excluindo quaisquer módulos carregados anteriormente na memória.

Se ocorrerem erros durante essa fase, uma chamada MQDISC explícita falhará com CompCode MQCC\_FAILED e o código de razão a seguir (erros não são destacados em solicitações de desconexão implícitas):

**MQRC\_API\_EXIT\_TERM\_ERROR**

Um erro foi encontrado durante o fechamento do ambiente de execução para uma função de saída de API. A saída não deve retornar nenhuma falha do MQDISC antes ou após as chamadas de função de saída da API MQ\_TERM\*..

**Saídas de API em clientes**

Um cliente usa a saída PreConnect para modificar o comportamento das chamada MQCONN e MQCONNX e não suporta as propriedades de saída de API..

**Saída do PreConnect**

Em um cliente, a saída PreConnect pode ser usada para consultar a definição de canal de um repositório central, como um servidor LDAP.

A saída PreConnect também pode modificar qualquer parâmetro ou todos os parâmetros em uma própria chamada MQCONN ou MQCONNX, por exemplo, o nome do gerenciador de filas.

No caso de aplicativos clientes, a saída PreConnect deve ser chamada antes da saída de API porque a saída de API MQCONN ou MQCONNX é chamada apenas uma vez que o nome do gerenciador de filas é conhecido e esse nome pode ser mudado pela saída PreConnect .

Observe que apenas a chamada MQCONN ou MQCONNX inicial chama a saída..

**Propriedades de saída da API.**

Em um servidor, saídas de API podem registrar uma estrutura MQXEPO no momento da inicialização. A estrutura MQXEPO contém o campo ExitProperties que detalha o grupo de propriedades no qual a saída está interessada.. Isso tem o efeito de gerar uma manipulação de propriedade de mensagem separada que a saída pode manipular separadamente de qualquer manipulação de propriedades de mensagem do aplicativo

Em um cliente, as propriedades de saída da API não são suportadas Se for feita uma tentativa de registrar um nome do grupo de propriedades em um cliente, a função falhará com um código de razão MQRC\_EXIT\_PROPS\_NOT\_SUPPORTED.

**Restauração-MQ\_BACK\_EXIT**

MQ\_BACK\_EXIT fornece uma função de saída de restauração para executar *antes* e *após* o processamento de restauração. Use o identificador de função MQXF\_BACK com as razões de saída

MQXR\_BEFORE e MQXR\_AFTER para registrar as funções de saída de chamada de restauração *antes* e *após* ..

A interface para esta função é:

```
MQ_BACK_EXIT (&ExitParms, &ExitContext, &Hconn, &CompCode, &Reason)
```

em que os parâmetros são:

**ExitParms (MQAXP)-entrada/saída**

Estrutura do parâmetro de saída

**ExitContext (MQAXC)-entrada/saída**

Estrutura do contexto de saída

**Hconn (MQHCONN)-entrada**

Manipulação de conexões.

**CompCode (MQLONG)-entrada/saída.**

Código de conclusão, valores válidos para os quais são:

**MQCC\_OK**

Indica conclusão bem-sucedida.

**MQCC\_WARNING**

Conclusão parcial.

**MQCC\_FAILED**

Chamada com falha

**Razão (MQLONG)-entrada/saída**

Código de razão qualificando o código de conclusão.

Se o código de conclusão for MQCC\_OK, o único valor válido será:

**MQRC\_NONE**

(0, X'000') Nenhuma razão a relatar.

Se o código de conclusão for MQCC\_FAILED ou MQCC\_WARNING, a função de saída poderá configurar o campo de código de razão como qualquer valor MQRC\_\* válido.

## Chamada de linguagem C

O gerenciador de filas define logicamente as variáveis a seguir:

```
MQAXP    ExitParms;      /* Exit parameter structure */
MQAXC    ExitContext;    /* Exit context structure */
MQHCONN  Hconn;         /* Connection handle */
MQLONG   CompCode;      /* Completion code */
MQLONG   Reason;        /* Reason code qualifying completion code */
```

O gerenciador de filas chama logicamente a saída da seguinte forma:

```
MQ_BACK_EXIT (&ExitParms, &ExitContext, &Hconn, &CompCode, &Reason);
```

Sua saída deve corresponder ao seguinte protótipo de função C:

```
void MQENTRY MQ_BACK_EXIT (
PMQAXP    pExitParms,    /* Address of exit parameter structure */
PMQAXC    pExitContext,  /* Address of exit context structure */
PMQHCONN  pHconn,       /* Address of connection handle */
PMQLONG   pCompCode,    /* Address of completion code */
PMQLONG   pReason);     /* Address of reason code qualifying completion
                          code */
```

## Início-MQ\_BEGIN\_EXIT

MQ\_BEGIN\_EXIT fornece uma função de saída inicial para executar *antes* e *após* o processamento de chamada MQBEGIN. Use o identificador de função MQXF\_BEGIN com as razões de saída MQXR\_BEFORE e MQXR\_AFTER para registrar *antes* e *após* funções de saída de chamada MQBEGIN.

A interface para esta função é:

```
MQ_BEGIN_EXIT (&ExitParms, &ExitContext, &Hconn, &pBeginOptions, &CompCode,  
              &Reason)
```

em que os parâmetros são:

### ExitParms (MQAXP)-entrada/saída

Estrutura do parâmetro de saída

### ExitContext (MQAXC)-entrada/saída

Estrutura do contexto de saída

### Hconn (MQHCONN)-entrada

Manipulação de conexões.

### pBeginOpções (PMQBO)-entrada/saída

Ponteiro para iniciar as opções

### CompCode (MQLONG)-entrada/saída.

Código de conclusão, valores válidos para os quais são:

#### MQCC\_OK

Indica conclusão bem-sucedida.

#### MQCC\_WARNING

Conclusão parcial.

#### MQCC\_FAILED

Chamada com falha

### Razão (MQLONG)-entrada/saída

Código de razão qualificando o código de conclusão.

Se o código de conclusão for MQCC\_OK, o único valor válido será:

#### MQRC\_NONE

(0, X'000') Nenhuma razão a relatar.

Se o código de conclusão for MQCC\_FAILED ou MQCC\_WARNING, a função de saída poderá configurar o campo de código de razão como qualquer valor MQRC\_\* válido.

## Chamada de linguagem C

O gerenciador de filas define logicamente as variáveis a seguir:

```
MQAXP    ExitParms;      /* Exit parameter structure */  
MQAXC    ExitContext;   /* Exit context structure */  
MQHCONN  Hconn;         /* Connection handle */  
PMQBO    pBeginOptions; /* Ptr to begin options */  
MQLONG   CompCode;     /* Completion code */  
MQLONG   Reason;       /* Reason code qualifying completion code */
```

O gerenciador de filas chama logicamente a saída da seguinte forma:

```
MQ_BEGIN_EXIT (&ExitParms, &ExitContext, &Hconn, &pBeginOptions, &CompCode,  
              &Reason);
```

Sua saída deve corresponder ao seguinte protótipo de função C:

```
void MQENTRY MQ_BEGIN_EXIT (
```



```

PMQAXP    pExitParms,      /* Address of exit parameter structure */
PMQAXC    pExitContext,  /* Address of exit context structure */
PMQHCONN  pHConn,       /* Address of connection handle */
PPMQBO    ppBeginOptions, /* Address of ptr to begin options */
PMQLONG   pCompCode,    /* Address of completion code */
PMQLONG   pReason;      /* Address of reason code qualifying completion
                        code */

```

### Retorno de chamada-MQ\_CALLBACK\_EXIT

MQ\_CALLBACK\_EXIT fornece uma função de saída para executar *antes* e *após* processamento de retorno de chamada. Use o identificador de função MQXF\_CALLBACK com as razões de saída MQXR\_BEFORE e MQXR\_AFTER para registrar *antes* e *após* funções de saída de chamada de retorno.

A interface para esta função é:

```

MQ_CALLBACK_EXIT (&ExitParms, &ExitContext, &Hconn, &pMsgDesc, &pGetMsgOpts,
                 &pBuffer, &pMQCBCContext)

```

em que os parâmetros são:

#### ExitParms (MQAXP)-entrada/saída

Estrutura do parâmetro de saída

#### ExitContext (MQAXC)-entrada/saída

Estrutura do contexto de saída

#### Hconn (MQHCONN)-entrada / saída

Identificador de conexões

#### pMsgDesc

Descritor de Mensagens

#### pGetMsgOpts

Opções que controlam a ação do MQGET

#### pBuffer

Área para conter os dados da mensagem

#### pMQCBCContext

Dados de contexto para o retorno de chamada

## Chamada de linguagem C

O gerenciador de filas define logicamente as variáveis a seguir:

```

MQAXP    ExitParms;      /* Exit parameter structure */
MQAXC    ExitContext;   /* Exit context structure */
MQHCONN  Hconn;         /* Connection handle */
PMQMD    pMsgDesc;      /* Message descriptor */
PMQGM0   pGetMsgOpts;   /* Options that define the operation of the consumer */
PMQVOID  pBuffer;       /* Area to contain the message data */
PMQCBC   pContext;      /* Context data for the callback */

```

O gerenciador de filas chama logicamente a saída da seguinte forma:

```

MQ_SUBRQ_EXIT (&ExitParms, &ExitContext, &Hconn, &pMsgDesc, &pGetMsgOpts, &pBuffer,
              &pContext);

```

Sua saída deve corresponder ao seguinte protótipo de função C:

```

void MQENTRY MQ_CALLBACK_EXIT (
PMQAXP    pExitParms;   /* Exit parameter structure */
PMQAXC    pExitContext; /* Exit context structure */
PMQHCONN  pHConn;       /* Connection handle */
PPMQMD    ppMsgDesc;    /* Message descriptor */
PPMQGM0   ppGetMsgOpts; /* Options that define the operation of the consumer */

```

```
PPMQVOID ppBuffer;      /* Area to contain the message data */
PPMQCBC  ppContext;     /* Context data for the callback */
```

## Observações de Uso

1. A saída de Retorno de Chamada é chamada antes do consumidor ser chamado e após a função do consumidor do consumidor ser concluída. Embora as estruturas MQMD e MQGMO sejam alteráveis, a mudança dos valores na saída anterior não redireciona a recuperação de uma mensagem da fila, pois a mensagem já foi removida da fila para ser entregue à função do consumidor

### Gerenciar funções de retorno de chamada-MQ\_CB\_EXIT

MQ\_CB\_EXIT fornece uma função de saída para executar *antes* e *após* a chamada MQCB. Use o identificador de função MQXF\_CB com as razões de saída MQXR\_BEFORE e MQXR\_AFTER para registrar *antes* e *após* funções de saída de chamada MQCB.

A interface para esta função é:

```
MQ_CB_EXIT (&ExitParms, &ExitContext, &Hconn, &Operation, &pCallbackDesc,
            &Hobj, &pMsgDesc, &pGetMsgOpts, &CompCode, &Reason)
```

em que os parâmetros são:

#### ExitParms (MQAXP)-entrada/saída

Estrutura do parâmetro de saída

#### ExitContext (MQAXC)-entrada/saída

Estrutura do contexto de saída

#### Hconn (MQHCONN)-entrada / saída

Identificador de conexões

#### Operação (MQLONG)-entrada/saída.

Valor da operação

#### pCallbackDesc (PMQCBD)-entrada / saída

Descritor de retorno de chamada

#### Hobj (MQHOBJ)-entrada / saída

Manipulação de objetos

#### pMsgDesc (PMQMD)-entrada / saída

Descritor de Mensagens

#### pGetMsgOpts (PMQGMO)-entrada / saída

Opções que controlam a ação de MQCB

#### CompCode (MQLONG)-entrada/saída.

Código de conclusão

#### Razão (MQLONG)-entrada/saída

Código de razão que qualifica CompCode

## Chamada de linguagem C

O gerenciador de filas define logicamente as variáveis a seguir:

```
MQAXP      ExitParms;      /* Exit parameter structure */
MQAXC      ExitContext;    /* Exit context structure */
MQHCONN    Hconn;         /* Connection handle */
MQLONG     Operation;     /* Operation value. */
MQCBD      pMsgDesc;      /* Callback descriptor. */
MQHOBJ     Hobj;         /* Object handle. */
PMQMD      pMsgDesc;      /* Message descriptor */
PMQGMO     pGetMsgOpts;   /* Options that define the operation of the consumer */
PMQLONG    CompCode;     /* Completion code. */
PMQLONG    Reason;       /* Reason code qualifying CompCode. */
```

O gerenciador de filas chama logicamente a saída da seguinte forma:

```
MQ_CB_EXIT (&ExitParms, &ExitContext, &Hconn, &Operation, &Hobj, &pMsgDesc,  
            &pGetMsgOpts, &CompCode, &Reason);
```

Sua saída deve corresponder ao seguinte protótipo de função C:

```
void MQENTRY MQ_CB_EXIT (  
PMQAXP    pExitParms;    /* Exit parameter structure */  
PMQAXC    pExitContext;  /* Exit context structure */  
PMQHCONN  pHconn;        /* Connection handle */  
PMQLONG   pOperation;    /* Callback operation */  
PMQHOBJS  pHobj;         /* Object handle */  
PPMQMD    pMsgDesc;      /* Message descriptor */  
PPMQGMO   ppGetMsgOpts;  /* Options that control the action of MQCB */  
PMQLONG   pCompCode;     /* Completion code */  
PMQLONG   pReason;       /* Reason code qualifying CompCode */
```

### **Fechar-MQ\_CLOSE\_EXIT**

MQ\_CLOSE\_EXIT fornece uma função de saída fechada para executar *antes* e *após* o processamento de chamada MQCLOSE. Use o identificador de função MQXF\_CLOSE com as razões de saída MQXR\_BEFORE e MQXR\_AFTER para registrar *antes* e *após* as funções de saída de chamada MQCLOSE.

A interface para esta função é:

```
MQ_CLOSE_EXIT (&ExitParms, &ExitContext, &Hconn, &pHobj,  
              &Options, &CompCode, &Reason)
```

em que os parâmetros são:

#### **ExitParms (MQAXP)-entrada/saída**

Estrutura do parâmetro de saída

#### **ExitContext (MQAXC)-entrada/saída**

Estrutura do contexto de saída

#### **Hconn (MQHCONN)-entrada**

Manipulação de conexões.

#### **pHobj (PMQHOBJS)-entrada de**

Ponteiro para a manipulação de objetos

#### **Opções (MQLONG)-entrada/saída.**

Feche as opções

#### **CompCode (MQLONG)-entrada/saída.**

Código de conclusão, valores válidos para os quais são:

##### **MQCC\_OK**

Indica conclusão bem-sucedida.

##### **MQCC\_FAILED**

Chamada com falha

#### **Razão (MQLONG)-entrada/saída**

Código de razão qualificando o código de conclusão.

Se o código de conclusão for MQCC\_OK, o único valor válido será:

##### **MQRC\_NONE**

(0, X'000') Nenhuma razão a relatar.

Se o código de conclusão for MQCC\_FAILED, a função de saída poderá configurar o campo de código de razão para qualquer valor MQRC\_ \* válido.

## Chamada de linguagem C

O gerenciador de filas define logicamente as variáveis a seguir:

```
MQAXP      ExitParms;      /* Exit parameter structure */
MQAXC      ExitContext;    /* Exit context structure */
MQHCONN    Hconn;         /* Connection handle */
PMQHOBJS   pHobj;        /* Ptr to object handle */
MQLONG     Options;       /* Close options */
MQLONG     CompCode;      /* Completion code */
MQLONG     Reason;       /* Reason code */
```

O gerenciador de filas chama logicamente a saída da seguinte forma:

```
MQ_CLOSE_EXIT (&ExitParms, &ExitContext,&Hconn, &pHobj, &Options,
               &CompCode, &Reason);
```

Sua saída deve corresponder ao seguinte protótipo de função C:

```
void MQENTRY MQ_CLOSE_EXIT (
PMQAXP      pExitParms,    /* Address of exit parameter structure */
PMQAXC      pExitContext,  /* Address of exit context structure */
PMQHCONN    pHconn,       /* Address of connection handle */
PPMHOBJS   ppHobj,        /* Address of ptr to object handle */
PMQLONG     pOptions,      /* Address of close options */
PMQLONG     pCompCode,     /* Address of completion code */
PMQLONG     pReason);     /* Address of reason code qualifying
                           completion code */
```

### Confirmação-MQ\_CMITS\_EXIT

MQ\_CMITS\_EXIT fornece uma função de saída de confirmação para executar *antes* e *após* processamento de confirmação. Use o identificador de função MQXF\_CMITS com as razões de saída MQXR\_BEFORE e MQXR\_AFTER para registrar *antes* e *depois* das funções de saída de chamada de confirmação..

Se uma operação de confirmação falhar, e a transação for restaurada, a chamada MQCMITS falhará com MQCC\_WARNING e MQRC\_BACKED\_OUT Esses códigos de retorno e de razão são transmitidos para qualquer *após* as funções de saída MQCMITS para fornecer às funções de saída uma indicação de que a unidade de trabalho foi restaurada.

A interface para esta função é:

```
MQ_CMITS_EXIT (&ExitParms, &ExitContext, &Hconn, &CompCode, &Reason)
```

em que os parâmetros são:

#### ExitParms (MQAXP)-entrada/saída

Estrutura do parâmetro de saída

#### ExitContext (MQAXC)-entrada/saída

Estrutura do contexto de saída

#### Hconn (MQHCONN)-entrada

Manipulação de conexões.

#### CompCode (MQLONG)-entrada/saída.

Código de conclusão, valores válidos para os quais são:

##### MQCC\_OK

Indica conclusão bem-sucedida.

##### MQCC\_WARNING

Conclusão parcial.

##### MQCC\_FAILED

Chamada com falha

## Razão (MQLONG)-entrada/saída

Código de razão qualificando o código de conclusão.

Se o código de conclusão for MQCC\_OK, o único valor válido será:

### MQRC\_NONE

(0, X'000') Nenhuma razão a relatar.

Se o código de conclusão for MQCC\_FAILED ou MQCC\_WARNING, a função de saída poderá configurar o campo de código de razão como qualquer valor MQRC\_\* válido.

## Chamada de linguagem C

O gerenciador de filas define logicamente as variáveis a seguir:

```
MQAXP    ExitParms;      /* Exit parameter structure */
MQAXC    ExitContext;    /* Exit context structure */
MQHCONN  Hconn;         /* Connection handle */
MQLONG   CompCode;      /* Completion code */
MQLONG   Reason;        /* Reason code qualifying completion code */
```

O gerenciador de filas chama logicamente a saída da seguinte forma:

```
MQ_CMITY_EXIT (&ExitParms, &ExitContext, &Hconn, &CompCode, &Reason);
```

Sua saída deve corresponder ao seguinte protótipo de função C:

```
void MQENTRY MQ_CMITY_EXIT (
PMQAXP    pExitParms,      /* Address of exit parameter structure */
PMQAXC    pExitContext,    /* Address of exit context structure */
PMQHCONN  pHconn,         /* Address of connection handle */
PMQLONG   pCompCode,      /* Address of completion code */
PMQLONG   pReason);       /* Address of reason code qualifying completion
                           code */
```

## Observações de Uso

1. A interface da função MQ\_GET\_EXIT descrita aqui é usada para a função de saída MQXF\_GET e a função de entrada [“MQXF\\_DATA\\_CONV\\_ON\\_GET”](#) na página 1648 .

Pontos de entrada separados são definidos para essas duas funções de saída, portanto, para interceptar *ambos* a chamada MQXEP deve ser usada duas vezes; para essa chamada, use o identificador de função MQXF\_GET.

Como a interface MQ\_GET\_EXIT é a mesma para MQXF\_GET e MQXF\_DATA\_CONV\_ON\_GET, uma única função de saída pode ser usada para ambos; o campo *Function* na estrutura MQAXP indica qual função de saída foi chamada. Como alternativa, a chamada MQXEP pode ser usada para registrar diferentes funções de saída para os dois casos.

### Conectar e conectar a extensão-MQ\_CONNX\_EXIT

MQ\_CONNX\_EXIT fornece função de saída de conexão para executar *antes* e *após* processamento MQCONN e função de saída de extensão de conexão para executar *antes* e *após* processamento MQCONNX.

A mesma interface, conforme descrito aqui, é chamada para as funções de saída de chamada MQCONN e MQCONNX.

Quando o agente do canal de mensagens (MCA) responde a uma conexão do cliente de entrada, o MCA pode se conectar e fazer várias chamadas API do IBM MQ antes que o estado do cliente seja totalmente conhecido. Essas chamadas API chamam as funções de saída da API com o MQAXC com base no próprio programa MCA (por exemplo, nos campos UserId e ConnectionName do MQAXC).

Quando o MCA responde a chamadas API do cliente de entrada subsequentes, a estrutura MQAXC é baseada no cliente de entrada, configurando os campos UserId e ConnectionName apropriadamente.

O nome do gerenciador de filas configurado pelo aplicativo em uma chamada MQCONN ou MQCONNX é transmitido para a chamada de conexão subjacente.. Qualquer tentativa feita por um *antes* de MQ\_CONNX\_EXIT para alterar o nome do gerenciador de fila não tem efeito

Use os identificadores de função MQXF\_CONN e MQXF\_CONNX com as razões de saída MQXR\_BEFORE e MQXR\_AFTER para registrar *antes* e *após* as funções de saída de chamada MQCONN e MQCONNX.

Uma saída MQ\_CONNX\_EXIT chamada pela razão MQXR\_BEFORE *não deve* emitir nenhuma IBM MQ chamada de API, pois o ambiente correto não foi configurado neste momento

Um MQ\_CONNX\_EXIT não pode chamar MQDISC de uma chamada de saída de API para a conexão para a qual ele está sendo chamado. Essa restrição é aplicável às saídas de API do cliente e do servidor.

A interface para MQCONN e MQCONNX é idêntica:

```
MQ_CONNX_EXIT (&ExitParms, &ExitContext, &pQMgrName, &pConnectOpts,  
&pHconn, &CompCode, &Reason);
```

em que os parâmetros são:

**ExitParms (MQAXP)-entrada/saída**

Estrutura do parâmetro de saída

**ExitContext (MQAXC)-entrada/saída**

Estrutura do contexto de saída

**pQMgrNome (PMQCHAR)-entrada**

Ponteiro para o nome do gerenciador de filas fornecido na chamada MQCONNX. A saída não deve alterar esse nome na chamada MQCONN ou MQCONNX

**pConnectOpts (PMQCNO)-entrada / saída**

Ponteiro para as opções que controlam a ação da chamadas MQCONNX.

Consulte “[MQCNO-Opções de conexão](#)” na página 324 para obter detalhes.

Para a função de saída MQXF\_CONN, pConnectOpts aponta para a estrutura de opções de conexão padrão (MQCNO\_DEFAULT).

**pHconn (PMQHCONN)-entrada**

Ponteiro para o identificador de conexão

**CompCode (MQLONG)-entrada/saída.**

Código de conclusão, valores válidos para os quais são:

**MQCC\_OK**

Indica conclusão bem-sucedida.

**MQCC\_WARNING**

Aviso (conclusão parcial)

**MQCC\_FAILED**

Chamada com falha

**Razão (MQLONG)-entrada/saída**

Código de razão qualificando o código de conclusão.

Se o código de conclusão for MQCC\_OK, o único valor válido será:

**MQRC\_NONE**

(0, X'000') Nenhuma razão a relatar.

Se o código de conclusão for MQCC\_FAILED ou MQCC\_WARNING, a função de saída poderá configurar o campo de código de razão como qualquer valor MQRC\_\* válido.

## Chamada de linguagem C

O gerenciador de filas define logicamente as variáveis a seguir:

```
MQAXP      ExitParms;      /* Exit parameter structure */
MQAXC      ExitContext;    /* Exit context structure */
PMQCHAR    pQMgrName;     /* Ptr to Queue manager name */
PMQCNO     pConnectOpts;  /* Ptr to Connection options */
PMQHCONN   pHconn;       /* Ptr to Connection handle */
MQLONG     CompCode;      /* Completion code */
MQLONG     Reason;       /* Reason code */
```

O gerenciador de filas chama logicamente a saída da seguinte forma:

```
MQ_CONNX_EXIT (&ExitParms, &ExitContext, &pQMgrName, &pConnectOps,
               &pHconn, &CompCode, &Reason);
```

Sua saída deve corresponder ao seguinte protótipo de função C:

```
void MQENTRY MQ_CONNX_EXIT (
PMQAXP      pExitParms,    /* Address of exit parameter structure */
PMQAXC      pExitContext,  /* Address of exit context structure */
PPMQCHAR    ppQMgrName,   /* Address of ptr to queue manager name */
PPMCNO      ppConnectOpts, /* Address of ptr to connection options */
PPMHCONN    ppHconn,      /* Address of ptr to connection handle */
PMQLONG     pCompCode,    /* Address of completion code */
PMQLONG     pReason);     /* Address of reason code qualifying
                           completion code */
```

## Observações de Uso

1. A interface de função MQ\_CONNX\_EXIT descrita aqui é usada para a chamada MQCONN e a chamada MQCONNX. Entretanto, pontos de entrada separados são definidos para essas duas chamadas. Para interceptar *ambas* as chamadas, a chamada MQXEP deve ser usada pelo menos duas vezes uma vez com o identificador de função MQXF\_CONN e novamente com MQXF\_CONNX.

Como a interface MQ\_CONNX\_EXIT é igual para MQCONN e MQCONNX, uma única função de saída pode ser usada para ambas as chamadas; o campo *Function* na estrutura MQAXP indica qual chamada está em andamento. Como alternativa, a chamada MQXEP pode ser usada para registrar diferentes funções de entrada para as duas chamadas.

2. Quando um agente do canal de mensagens (MCA) responde a uma conexão do cliente de entrada, o MCA pode emitir várias chamadas do MQ antes que o estado do cliente seja totalmente conhecido. Essas chamadas MQ resultam nas funções de saída de API sendo chamadas com a estrutura MQAXC contendo dados relacionados ao MCA e não ao cliente (por exemplo, identificador de usuário e nome de conexão). No entanto, quando o estado do cliente for totalmente conhecido, as chamadas subsequentes do MQ resultarão nas funções de saída de API sendo chamadas com os dados do cliente apropriados na estrutura MQAXC.
3. Todas as funções de saída MQXR\_BEFORE são chamadas antes que qualquer validação de parâmetro seja executada por um gerenciador de fila. Portanto, os parâmetros podem ser inválidos (incluindo ponteiros inválidos para os endereços de parâmetros).

A função MQ\_CONNX\_EXIT é chamada antes de qualquer verificação de autorização ser executada pelo gerenciador de filas..

4. A função de saída não deve mudar o nome do gerenciador de filas especificado na chamada MQCONN ou MQCONNX. Se o nome for alterado pela função de saída, os resultados serão indefinidos.
5. Uma função de saída MQXR\_BEFORE para MQ\_CONNX\_EXIT não pode emitir chamadas MQ diferentes de MQXEP.

## Retorno de chamada de controle-MQ\_CTL\_EXIT

MQ\_CTL\_EXIT fornece uma função de saída de pedido de assinatura para executar *antes* e *após* controlar o processamento de retorno de chamada. Use o identificador de função MQXF\_CTL com as razões de

saída MQXR\_BEFORE e MQXR\_AFTER para registrar *antes* e *após* as funções de saída de chamada de retorno de controle.

A interface para esta função é:

```
MQ_CTL_EXIT (&Hconn, &Operation, &ControlOpts, &CompCode, &Reason)
```

em que os parâmetros são:

**Hconn (MQHCONN)-entrada / saída**

Manipulação de conexões.

**Entrada / saída da operação (MQLONG)**

A operação sendo processada no retorno de chamada definido para a manipulação de objetos especificada

**ControlOpts (MQCTLO) de entrada / saída**

Opções que controlam a ação de MQCTL

**CompCode (MQLONG)-entrada/saída.**

Código de conclusão, valores válidos para os quais são:

**MQCC\_OK**

Indica conclusão bem-sucedida.

**MQCC\_WARNING**

Conclusão parcial.

**MQCC\_FAILED**

Chamada com falha

**Razão (MQLONG)-entrada/saída**

Código de razão qualificando o código de conclusão.

Se o código de conclusão for MQCC\_OK, o único valor válido será:

**MQRC\_NONE**

(0, X'000') Nenhuma razão a relatar.

Se o código de conclusão for MQCC\_FAILED ou MQCC\_WARNING, a função de saída poderá configurar o campo de código de razão como qualquer valor MQRC\_\* válido.

## Chamada de linguagem C

O gerenciador de filas define logicamente as variáveis a seguir:

```
MQHCONN  Hconn;          /* Connection handle */
MQLONG   Operation;     /* Operation being processed */
MQCTLO   ControlOpts;  /* Options that control the action of MQCTL */
MQLONG   CompCode;     /* Completion code */
MQLONG   Reason;       /* Reason code qualifying completion code */
```

O gerenciador de filas chama logicamente a saída da seguinte forma:

```
MQ_CTL_EXIT (&Hconn, &Operation, &ControlOpts, &CompCode, &Reason);
```

Sua saída deve corresponder ao seguinte protótipo de função C:

```
void MQENTRY MQ_CTL_EXIT (
PMQHCONN  pHconn;       /* Address of connection handle */
PMQLONG   pOperation;   /* Address of operation being processed */
PMQCTLO   pControlOpts; /* Address of options that control the action of MQCTL */
PMQLONG   pCompCode;    /* Address of completion code */
PMQLONG   pReason;     /* Address of reason code qualifying completion code */
```



## Desconexão-MQ\_DISC\_EXIT

MQ\_DISC\_EXIT fornece uma função de saída de desconexão para executar *antes* e *após* o processamento de saída MQDISC. Use o identificador de função MQXF\_DISC com as razões de saída MQXR\_BEFORE e MQXR\_AFTER para registrar *antes* e *após* as funções de saída de chamada MQDISC..

A interface para esta função é

```
MQ_DISC_EXIT (&ExitParms, &ExitContext, &pHconn,  
&CompCode, &Reason);
```

em que os parâmetros são:

### ExitParms (MQAXP)-entrada/saída

Estrutura do parâmetro de saída

### ExitContext (MQAXC)-entrada/saída

Estrutura do contexto de saída

### pHconn (PMQHCONN)-entrada

Ponteiro para o identificador de conexão

Para a chamada MQDISC anterior, o valor desse campo é um de:

- O identificador de conexão retornado na chamada MQCONN ou MQCONNX
- Zero, para ambientes em que um adaptador específico do ambiente foi conectado ao gerenciador de filas
- Um valor configurado por uma chamada de função de saída anterior

Para a chamada MQDISC após, o valor desse campo é zero ou um valor configurado por uma chamada de função de saída anterior.

### CompCode (MQLONG)-entrada/saída.

Código de conclusão, valores válidos para os quais são:

#### MQCC\_OK

Indica conclusão bem-sucedida.

#### MQCC\_WARNING

Conclusão parcial

#### MQCC\_FAILED

Chamada com falha

### Razão (MQLONG)-entrada/saída

Código de razão qualificando o código de conclusão.

Se o código de conclusão for MQCC\_OK, o único valor válido será:

#### MQRC\_NONE

(0, X'000') Nenhuma razão a relatar.

Se o código de conclusão for MQCC\_FAILED ou MQCC\_WARNING, a função de saída poderá configurar o campo de código de razão como qualquer valor MQRC\_\* válido.

## Chamada de linguagem C

O gerenciador de filas define logicamente as variáveis a seguir:

```
MQAXP          ExitParms;      /* Exit parameter structure */  
MQAXC          ExitContext;    /* Exit context structure */  
PMQHCONN      pHconn;         /* Ptr to Connection handle */  
MQLONG        CompCode;       /* Completion code */  
MQLONG        Reason;         /* Reason code */
```

O gerenciador de filas chama logicamente a saída da seguinte forma:

```
MQ_DISC_EXIT (&ExitParms, &ExitContext, &pHconn,  
              &CompCode, &Reason);
```

Sua saída deve corresponder ao seguinte protótipo de função C:

```
void MQENTRY MQ_DISC_EXIT (  
    PMQAXP      pExitParms,      /* Address of exit parameter structure */  
    PMQAXC      pExitContext,    /* Address of exit context structure */  
    PPMQHCONN   ppHconn,        /* Address of ptr to connection handle */  
    PMQLONG     pCompCode,       /* Address of completion code */  
    PMQLONG     pReason);       /* Address of reason code qualifying  
                                completion code */
```

### **Obter-MQ\_GET\_EXIT**

MQ\_GET\_EXIT fornece uma função get exit para executar *antes* e *após* processamento de chamada MQGET.

Há dois identificadores de função:

1. Use MQXF\_GET com as razões de saída MQXR\_BEFORE e MQXR\_AFTER para registrar *antes* e *após* funções de saída de chamada MQGET.
2. Consulte [“MQXF\\_DATA\\_CONV\\_ON\\_GET”](#) na página 1648 para obter informações sobre o uso do identificador de função MQXF\_DATA\_CONV\_ON\_GET..

A interface para esta função é:

```
MQ_GET_EXIT (&ExitParms, &ExitContext, &Hconn, &Hobj, &pMsgDesc,  
            &pGetMsgOpts, &BufferLength, &pBuffer, &pDataLength,  
            &CompCode, &Reason)
```

em que os parâmetros são:

#### **ExitParms (MQAXP)-entrada/saída**

Estrutura do parâmetro de saída

#### **ExitContext (MQAXC)-entrada/saída**

Estrutura do contexto de saída

#### **Hconn (MQHCONN)-entrada**

Manipulação de conexões.

#### **Hobj (MQHOBJ)-entrada / saída**

Manipulação de objetos.

#### **pMsgDesc (PMQMD)-entrada / saída**

Ponteiro para o descritor de mensagem

#### **pGetMsgOpts (PMQGMO)-entrada / saída**

Ponteiro para obter as opções de mensagem

#### **BufferLength (MQLONG)-entrada/saída**

Comprimento do buffer de mensagem..

#### **pBuffer (PMQBYTE)-entrada / saída**

Ponteiro para o buffer de mensagem

#### **pDataComprimento (PMQLONG)-entrada/saída.**

Ponteiro para campo de comprimento de dados.

#### **CompCode (MQLONG)-entrada/saída.**

Código de conclusão, valores válidos para os quais são:

##### **MQCC\_OK**

Indica conclusão bem-sucedida.

##### **MQCC\_WARNING**

Conclusão parcial.

## **MQCC\_FAILED**

Chamada com falha

### **Razão (MQLONG)-entrada/saída**

Código de razão qualificando o código de conclusão.

Se o código de conclusão for MQCC\_OK, o único valor válido será:

### **MQRC\_NONE**

(0, X'000') Nenhuma razão a relatar.

Se o código de conclusão for MQCC\_FAILED ou MQCC\_WARNING, a função de saída poderá configurar o campo de código de razão como qualquer valor MQRC\_\* válido.

## **Chamada de linguagem C**

O gerenciador de filas define logicamente as variáveis a seguir:

```
MQAXP      ExitParms;      /* Exit parameter structure */
MQAXC      ExitContext;    /* Exit context structure */
MQHCONN    Hconn;         /* Connection handle */
MQHOBJ     Hobj;          /* Object handle */
PMQMD      pMsgDesc;      /* Ptr to message descriptor */
PMQPMO     pGetMsgOpts;   /* Ptr to get message options */
MQLONG     BufferLength;   /* Message buffer length */
PMQBYTE    pBuffer;       /* Ptr to message buffer */
PMQLONG    pDataLength;   /* Ptr to data length field */
MQLONG     CompCode;      /* Completion code */
MQLONG     Reason;        /* Reason code */
```

O gerenciador de filas chama logicamente a saída da seguinte forma:

```
MQ_GET_EXIT (&ExitParms, &ExitContext, &Hconn, &Hobj, &pMsgDesc,
             &pGetMsgOpts, &BufferLength, &pBuffer, &pDataLength,
             &CompCode, &Reason)
```

Sua saída deve corresponder ao seguinte protótipo de função C:

```
void MQENTRY MQ_GET_EXIT (
PMQAXP      pExitParms,    /* Address of exit parameter structure */
PMQAXC      pExitContext,  /* Address of exit context structure */
PMQHCONN    pHconn,       /* Address of connection handle */
PMQHOBJ     pHobj,        /* Address of object handle */
PPMQMD      ppMsgDesc,    /* Address of ptr to message descriptor */
PPMQGMO     ppGetMsgOpts, /* Address of ptr to get message options */
PMQLONG     pBufferLength, /* Address of message buffer length */
PPMQBYTE    ppBuffer,     /* Address of ptr to message buffer */
PPMQLONG    ppDataLength, /* Address of ptr to data length field */
PMQLONG     pCompCode,    /* Address of completion code */
PMQLONG     pReason);     /* Address of reason code qualifying
                             completion code */
```

## **Observações de Uso**

1. A interface da função MQ\_GET\_EXIT descrita aqui é usada para a função de saída MQXF\_GET e a função de entrada [“MQXF\\_DATA\\_CONV\\_ON\\_GET”](#) na página 1648 .

Pontos de entrada separados são definidos para essas duas funções de saída, portanto, para interceptar *ambos* a chamada MQXEP deve ser usada duas vezes; para essa chamada, use o identificador de função MQXF\_GET.

Como a interface MQ\_GET\_EXIT é a mesma para MQXF\_GET e MQXF\_DATA\_CONV\_ON\_GET, uma única função de saída pode ser usada para ambos; o campo *Function* na estrutura [MQAXP](#) indica qual função de saída foi chamada. Como alternativa, a chamada MQXEP pode ser usada para registrar diferentes funções de saída para os dois casos.

## ***MQXF\_DATA\_CONV\_ON\_GET***

O identificador da função `MQXF_DATA_CONV_ON_GET` é usado com `MQ_GET_EXIT`.

Consulte `MQ_GET_EXIT` para obter informações sobre a interface para esta chamada e uma declaração de linguagem C de amostra

### **Observações de Uso**

Se registrado, esse ponto de entrada será chamado quando as mensagens chegarem ao aplicativo, mas antes que qualquer conversão de dados tenha ocorrido. Isso pode ser útil se a saída de API precisar executar o processamento, como decriptografia ou descompactação, antes que a mensagem seja transmitida para a conversão de dados. A saída pode, se necessário, fazer com que a conversão de dados seja ignorada retornando `MQXCC_SUPPRESS_FUNCTION`; para obter informações adicionais, consulte a estrutura `MQAXP`.

O registro para esse ponto de entrada em um cliente tem o efeito de fazer com que a conversão de dados seja executada localmente na máquina cliente. Para a operação correta, pode ser necessário, portanto, instalar as saídas de conversão do aplicativo no cliente. Observe que `MQXF_DATA_CONV_ON_GET` também é usado para consumo assíncrono.

Ao usar a chamada `MQ_GET_EXIT`, use `MQXF_DATA_CONV_ON_GET`, com razão de saída `MQXR_BEFORE`, para registrar uma função de saída de conversão de dados `MQGET` antes.

Não há nenhuma função de saída `MQXR_AFTER` para `MQXF_DATA_CONV_ON_GET`; a função de saída `MQXR_AFTER` para `MQXF_GET` fornece a capacidade necessária para processamento de saída após conversão de dados.

Pontos de entrada separados são definidos para a chamada `MQ_GET_EXIT`, portanto, para interceptar ambas as funções de saída, a chamada `MQXEP` deve ser usada duas vezes; para essa chamada, use o identificador de função `MQXF_DATA_CONV_ON_GET`.

Como a interface `MQ_GET_EXIT` é a mesma para `MQXF_GET` e `MQXF_DATA_CONV_ON_GET`, uma única função de saída pode ser usada para ambos; o campo *Function* na estrutura `MQAXP` indica qual função de saída foi chamada. Como alternativa, a chamada `MQXEP` pode ser usada para registrar diferentes funções de saída para os dois casos.

### ***Inicialização-MQ\_INIT\_EXIT***

`MQ_INIT_EXIT` fornece a inicialização do nível de conexão, indicada pela configuração de `ExitReason` em `MQAXP` para `MQXR_CONNECTION`.

Durante a inicialização, observe o seguinte:

- A função `MQ_INIT_EXIT` chama `MQXEP` para registrar os verbos da API IBM MQ e os pontos `ENTRY` e `EXIT` nos quais ela está interessada.
- As saídas não precisam interceptar todos os verbos da API do IBM MQ. As funções de saída serão chamadas somente se um interesse tiver sido registrado.
- O armazenamento que deve ser usado pela saída pode ser adquirido ao inicializá-lo.
- Se uma chamada para essa função falhar, a chamada `MQCONN` ou `MQCONNX` que a chamou também falhará com um `CompCode` e `Motivo` que dependem do valor do campo `ExitResponse` em `MQAXP`.
- Uma saída `MQ_INIT_EXIT` não deve emitir chamadas API IBM MQ, porque o ambiente correto não foi configurado neste momento.
- Se um `MQ_INIT_EXIT` falhar com `MQXCC_FAILED`, o gerenciador de filas retornará da chamada `MQCONN` ou `MQCONNX` que o chamou com `MQCC_FAILED` e `MQRC_API_EXIT_ERROR`.
- Se o gerenciador de filas encontrar um erro ao inicializar o ambiente de execução da função de saída da API antes de chamar o primeiro `MQ_INIT_EXIT`, o gerenciador de filas retornará da chamada `MQCONN` ou `MQCONNX` que chamou `MQ_INIT_EXIT` com `MQCC_FAILED` e `MQRC_API_EXIT_INIT_ERROR`.

A interface para `MQ_INIT_EXIT` é:

```
MQ_INIT_EXIT (&ExitParms, &ExitContext, &CompCode, &Reason)
```

em que os parâmetros são:

**ExitParms (MQAXP)-entrada/saída**

Estrutura do parâmetro de saída

**ExitContext (MQAXC)-entrada/saída**

Estrutura do contexto de saída

**CompCode (MQLONG)-entrada/saída.**

Ponteiro para o código de conclusão, os valores válidos para os quais são:

**MQCC\_OK**

Indica conclusão bem-sucedida.

**MQCC\_WARNING**

Conclusão parcial.

**MQCC\_FAILED**

Chamada com falha

**Razão (MQLONG)-entrada/saída**

Ponteiro para código de razão que qualifica o código de conclusão.

Se o código de conclusão for MQCC\_OK, o único valor válido será:

**MQRC\_NONE**

(0, X'000') Nenhuma razão a relatar.

Se o código de conclusão for MQCC\_FAILED ou MQCC\_WARNING, a função de saída poderá configurar o campo de código de razão como qualquer valor MQRC\_\* válido.

O CompCode e Reason retornados para o aplicativo dependem do valor do campo ExitResponse em MQAXP.

## Chamada de linguagem C

O gerenciador de filas define logicamente as variáveis a seguir:

```
MQAXP      ExitParms;      /* Exit parameter structure */
MQAXC      ExitContext;    /* Exit context structure */
MQLONG     CompCode;       /* Completion code */
MQLONG     Reason;         /* Reason code */
```

O gerenciador de filas chama logicamente a saída da seguinte forma:

```
MQ_INIT_EXIT (&ExitParms, &ExitContext, &CompCode, &Reason)
```

Sua saída deve corresponder ao seguinte protótipo de função C:

```
void MQENTRY MQ_INIT_EXIT (
PMQAXP      pExitParms,    /* Address of exit parameter structure */
PMQAXC      pExitContext,  /* Address of exit context structure */
PMQLONG     pCompCode,     /* Address of completion code */
PMQLONG     pReason);     /* Address of reason code qualifying
                             completion code */
```

## Observações de Uso

1. A função MQ\_INIT\_EXIT pode emitir a chamada MQXEP para registrar os endereços das funções de saída para as chamadas específicas do MQ a serem interceptadas.. Não é necessário interceptar todas as chamadas do MQ ou interceptar as chamadas MQXR\_BEFORE e MQXR\_AFTER. Por exemplo, um conjunto de saída poderia optar por interceptar apenas a chamada MQXR\_BEFORE de MQPUT

2. O armazenamento que deve ser usado por funções de saída no conjunto de saídas pode ser adquirido pela função MQ\_INIT\_EXIT.. Como alternativa, as funções de saída podem adquirir armazenamento quando elas são chamadas, conforme e quando necessário. No entanto, todo o armazenamento deve ser liberado antes do conjunto de saída ser finalizado; a função MQ\_TERM\_EXIT pode liberar o armazenamento ou uma função de saída chamada anteriormente.
3. Se MQ\_INIT\_EXIT retornar MQXCC\_FAILED no campo ExitResponse de MQAXP ou falhar de alguma outra maneira, a chamada MQCONN ou MQCONNX que fez com que MQ\_INIT\_EXIT fosse chamado também falhará, com os parâmetros **CompCode** e **Reason** configurados para valores apropriados.
4. Uma função MQ\_INIT\_EXIT não pode emitir chamadas MQ diferentes de MQXEP.

### **Consulta-MQ\_INQ\_EXIT**

MQ\_INQ\_EXIT fornece uma função de saída de consulta para executar *antes* e *após* o processamento de chamada MQINQ. Use o identificador de função MQXF\_INQ com as razões de saída MQXR\_BEFORE e MQXR\_AFTER para registrar *antes* e *após* funções de saída de chamada MQINQ.

A interface para esta função é:

```
MQ_INQ_EXIT (&ExitParms, &ExitContext, &Hconn, &Hobj, &SelectorCount,
            &pSelectors, &IntAttrCount, &pIntAttrs, &CharAttrLength,
            &pCharAttrs, &CompCode, &Reason)
```

em que os parâmetros são:

#### **ExitParms (MQAXP)-entrada/saída**

Estrutura do parâmetro de saída

#### **ExitContext (MQAXC)-entrada/saída**

Estrutura do contexto de saída

#### **Hconn (MQHCONN)-entrada**

Manipulação de conexões.

#### **Hobj (MQHOBJ)-entrada**

Manipulação de objetos.

#### **SelectorCount (MQLONG)-entrada**

Contagem de seletores.

#### **pSelectors (PMQLONG)-entrada / saída**

Ponteiro para a matriz de valores do seletor

#### **IntAttrContagem (MQLONG)-entrada**

Contagem de atributos de número inteiro.

#### **Attrs pInt(PMQLONG)-entrada/saída**

Ponteiro para matriz de valores de atributo de número inteiro.

#### **CharAttrComprimento (MQLONG)-entrada/saída.**

Comprimento da matriz de atributos de caracteres

#### **pCharAttrs (PMQCHAR)-entrada/saída**

Ponteiro para matriz de atributos de caracteres.

#### **CompCode (MQLONG)-entrada/saída.**

Código de conclusão, valores válidos para os quais são:

##### **MQCC\_OK**

Indica conclusão bem-sucedida.

##### **MQCC\_WARNING**

Conclusão parcial.

##### **MQCC\_FAILED**

Chamada com falha

#### **Razão (MQLONG)-entrada/saída**

Código de razão qualificando o código de conclusão.

Se o código de conclusão for MQCC\_OK, o único valor válido será:

### **MQRC\_NONE**

(0, X'000') Nenhuma razão a relatar.

Se o código de conclusão for MQCC\_FAILED ou MQCC\_WARNING, a função de saída poderá configurar o campo de código de razão como qualquer valor MQRC\_\* válido.

## **Chamada de linguagem C**

O gerenciador de filas define logicamente as variáveis a seguir:

```
MQAXP    ExitParms;        /* Exit parameter structure */
MQAXC    ExitContext;     /* Exit context structure */
MQHCONN  Hconn;          /* Connection handle */
MQHOBJ   Hobj;           /* Object handle */
MQLONG   SelectorCount;  /* Count of selectors */
PMQLONG  pSelectors;     /* Ptr to array of attribute selectors */
MQLONG   IntAttrCount;   /* Count of integer attributes */
PMQLONG  pIntAttrs;     /* Ptr to array of integer attributes */
MQLONG   CharAttrLength; /* Length of char attributes array */
PMQCHAR  pCharAttrs;    /* Ptr to character attributes */
MQLONG   CompCode;      /* Completion code */
MQLONG   Reason;        /* Reason code qualifying completion code */
```

O gerenciador de filas chama logicamente a saída da seguinte forma:

```
MQ_INQ_EXIT (&ExitParms, &ExitContext, &Hconn, &Hobj, &SelectorCount,
             &pSelectors, &IntAttrCount, &pIntAttrs, &CharAttrLength,
             &pCharAttrs, &CompCode, &Reason)
```

Sua saída deve corresponder ao seguinte protótipo de função C:

```
void MQENTRY MQ_INQ_EXIT (
PMQAXP    pExitParms,        /* Address of exit parameter structure */
PMQAXC    pExitContext,     /* Address of exit context structure */
PMQHCONN  pHconn,          /* Address of connection handle */
PMQHOBJ   pHobj,           /* Address of object handle */
PMQLONG   pSelectorCount,  /* Address of selector count */
PPMQLONG  ppSelectors,     /* Address of ptr to array of selectors */
PMQLONG   pIntAttrCount;   /* Address of count of integer attributes */
PPMQLONG  ppIntAttrs,     /* Address of ptr to array of integer attributes */
PMQLONG   pCharAttrLength, /* Address of character attribute length */
PPMQCHAR  ppCharAttrs,    /* Address of ptr to character attributes array */
PMQLONG   pCompCode,      /* Address of completion code */
PMQLONG   pReason);       /* Address of reason code qualifying completion
                           code */
```

### **Aberto-MQ\_OPEN\_EXIT**

MQ\_OPEN\_EXIT fornece uma função de saída aberta para executar *antes* e *após* o processamento de chamada MQOPEN.. Use o identificador de função MQXF\_OPEN com as razões de saída MQXR\_BEFORE e MQXR\_AFTER para registrar *antes* e *após* as funções de saída de chamada MQOPEN.

A interface para esta função é

```
MQ_OPEN_EXIT (&ExitParms, &ExitContext, &Hconn, &pObjDesc, &Options,
              &pHobj, &CompCode, &Reason)
```

em que os parâmetros são:

#### **ExitParms (MQAXP)-entrada/saída**

Estrutura do parâmetro de saída

#### **ExitContext (MQAXC)-entrada/saída**

Estrutura do contexto de saída

**Hconn (MQHCONN)-entrada**

Manipulação de conexões.

**pObjDesc (PMQOD)-entrada / saída**

Ponteiro para o descritor de objeto

**Opções (MQLONG)-entrada/saída.**

Abrir opções.

**pHobj (PMQHOBj)-entrada de**

Ponteiro para a manipulação de objetos

**CompCode (MQLONG)-entrada/saída.**

Código de conclusão, valores válidos para os quais são:

**MQCC\_OK**

Indica conclusão bem-sucedida.

**MQCC\_WARNING**

Conclusão parcial

**MQCC\_FAILED**

Chamada com falha

**Razão (MQLONG)-entrada/saída**

Código de razão qualificando o código de conclusão.

Se o código de conclusão for MQCC\_OK, o único valor válido será:

**MQRC\_NONE**

(0, X'000') Nenhuma razão a relatar.

Se o código de conclusão for MQCC\_FAILED ou MQCC\_WARNING, a função de saída poderá configurar o campo de código de razão como qualquer valor MQRC\_\* válido.

**Chamada de linguagem C**

O gerenciador de filas define logicamente as variáveis a seguir:

```
MQAXP      ExitParms;      /* Exit parameter structure */
MQAXC      ExitContext;    /* Exit context structure */
MQHCONN    Hconn;         /* Connection handle */
PMQOD      pObjDesc;      /* Ptr to object descriptor */
MQLONG     Options;       /* Open options */
PMQHOBj    pHobj;        /* Ptr to object handle */
MQLONG     CompCode;      /* Completion code */
MQLONG     Reason;       /* Reason code */
```

O gerenciador de filas chama logicamente a saída da seguinte forma:

```
MQ_OPEN_EXIT (&ExitParms, &ExitContext, &Hconn, &pObjDesc, &Options,
              &pHobj, &CompCode, &Reason);
```

Sua saída deve corresponder ao seguinte protótipo de função C:

```
void MQENTRY MQ_OPEN_EXIT (
PMQAXP      pExitParms,    /* Address of exit parameter structure */
PMQAXC      pExitContext, /* Address of exit context structure */
PMQHCONN    pHconn,       /* Address of connection handle */
PPMQOD      ppObjDesc,    /* Address of ptr to object descriptor */
PMQLONG     pOptions,     /* Address of open options */
PPMQHOBj    ppHobj,       /* Address of ptr to object handle */
PMQLONG     pCompCode,    /* Address of completion code */
PMQLONG     pReason);     /* Address of reason code qualifying
                             completion code */
```



## Put-MQ\_PUT\_EXIT

MQ\_PUT\_EXIT fornece uma função de saída put para executar *antes* e *após* processamento de chamada MQPUT. Use o identificador de função MQXF\_PUT com as razões de saída MQXR\_BEFORE e MQXR\_AFTER para registrar *antes* e *após* as funções de saída de chamada MQPUT

A interface para esta função é:

```
MQ_PUT_EXIT (&ExitParms, &ExitContext, &Hconn, &Hobj, &pMsgDesc,  
            &pPutMsgOpts, &BufferLength, &pBuffer, &CompCode, &Reason)
```

em que os parâmetros são:

### ExitParms (MQAXP)-entrada/saída

Estrutura do parâmetro de saída

### ExitContext (MQAXC)-entrada/saída

Estrutura do contexto de saída

### Hconn (MQHCONN)-entrada

Manipulação de conexões.

### Hobj (MQHOBJ)-entrada / saída

Manipulação de objetos.

### pMsgDesc (PMQMD)-entrada / saída

Ponteiro para o descritor de mensagem

### pPutMsgOpts (PMQPMO)-entrada / saída

Ponteiro para colocar as opções de mensagem.

### BufferLength (MQLONG)-entrada/saída

Comprimento do buffer de mensagem..

### pBuffer (PMQBYTE)-entrada / saída

Ponteiro para o buffer de mensagem

### CompCode (MQLONG)-entrada/saída.

Código de conclusão, valores válidos para os quais são:

#### MQCC\_OK

Indica conclusão bem-sucedida.

#### MQCC\_WARNING

Conclusão parcial.

#### MQCC\_FAILED

Chamada com falha

### Razão (MQLONG)-entrada/saída

Código de razão qualificando o código de conclusão.

Se o código de conclusão for MQCC\_OK, o único valor válido será:

#### MQRC\_NONE

(0, X'000') Nenhuma razão a relatar.

Se o código de conclusão for MQCC\_FAILED ou MQCC\_WARNING, a função de saída poderá configurar o campo de código de razão como qualquer valor MQRC\_\* válido.

## Chamada de linguagem C

O gerenciador de filas define logicamente as variáveis a seguir:

```
MQAXP      ExitParms;      /* Exit parameter structure */  
MQAXC      ExitContext;    /* Exit context structure */  
MQHCONN    Hconn;         /* Connection handle */  
MQHOBJ     Hobj;          /* Object handle */  
PMQMD      pMsgDesc;      /* Ptr to message descriptor */  
PMQPMO     pPutMsgOpts;   /* Ptr to put message options */
```

```

MQLONG      BufferLength; /* Message buffer length */
PMQBYTE     pBuffer;     /* Ptr to message data */
MQLONG      CompCode;    /* Completion code */
MQLONG      Reason;      /* Reason code */

```

O gerenciador de filas chama logicamente a saída da seguinte forma:

```

MQ_PUT_EXIT (&ExitParms, &ExitContext, &Hconn, &Hobj, &pMsgDesc,
             &pPutMsgOpts, &BufferLength, &pBuffer, &CompCode, &Reason)

```

Sua saída deve corresponder ao seguinte protótipo de função C:

```

void MQENTRY MQ_PUT_EXIT (
PMQAXP      pExitParms, /* Address of exit parameter structure */
PMQAXC      pExitContext, /* Address of exit context structure */
MQHCONN     pHconn, /* Address of connection handle */
MQHOBJ      pHobj, /* Address of object handle */
PMQMD       ppMsgDesc, /* Address of ptr to message descriptor */
PMQPMO      ppPutMsgOpts, /* Address of ptr to put message options */
MQLONG      pBufferLength, /* Address of message buffer length */
PMQBYTE     ppBuffer, /* Address of ptr to message buffer */
MQLONG      pCompCode, /* Address of completion code */
MQLONG      pReason); /* Address of reason code qualifying
                        completion code */

```

## Observações de Uso

- As mensagens de relatório geradas pelo gerenciador de filas ignoram o processamento de chamada normal. Como resultado, essas mensagens não podem ser interceptadas pela função MQ\_PUT\_EXIT ou pela função MQPUT1. No entanto, as mensagens de relatório geradas pelo agente do canal de mensagens são processadas normalmente e, portanto, podem ser interceptadas pela função MQ\_PUT\_EXIT ou pela função MQ\_PUT1\_EXIT. Para ter certeza de interceptar todas as mensagens de relatório geradas pelo MCA, ambos MQ\_PUT\_EXIT e MQ\_PUT1\_EXIT devem ser usados.

### Put1 - MQ\_PUT1\_EXIT

MQ\_PUT1\_EXIT fornece uma função de saída *put one message only* para executar *antes e após* MQPUT1 processamento de chamada. Use o identificador de função MQXF\_PUT1 com motivos de saída MQXR\_BEFORE e MQXR\_AFTER para registrar *antes e após* MQPUT1 funções de saída de chamada.

A interface para esta função é:

```

MQ_PUT1_EXIT (&ExitParms, &ExitContext, &Hconn, &pObjDesc, &pMsgDesc,
             &pPutMsgOpts, &BufferLength, &pBuffer, &CompCode, &Reason)

```

em que os parâmetros são:

#### ExitParms (MQAXP)-entrada/saída

Estrutura do parâmetro de saída

#### ExitContext (MQAXC)-entrada/saída

Estrutura do contexto de saída

#### Hconn (MQHCONN)-entrada

Manipulação de conexões.

#### pObjDesc (PMQOD)-entrada / saída

Ponteiro para o descritor de objeto

#### pMsgDesc (PMQMD)-entrada / saída

Ponteiro para o descritor de mensagem

#### pPutMsgOpts (PMQPMO)-entrada / saída

Ponteiro para colocar as opções de mensagem.

#### BufferLength (MQLONG)-entrada/saída

Comprimento do buffer de mensagem..

### **pBuffer (PMQBYTE)-entrada / saída**

Ponteiro para o buffer de mensagem

### **CompCode (MQLONG)-entrada/saída.**

Código de conclusão, valores válidos para os quais são:

#### **MQCC\_OK**

Indica conclusão bem-sucedida.

#### **MQCC\_WARNING**

Conclusão parcial.

#### **MQCC\_FAILED**

Chamada com falha

### **Razão (MQLONG)-entrada/saída**

Código de razão qualificando o código de conclusão.

Se o código de conclusão for MQCC\_OK, o único valor válido será:

#### **MQRC\_NONE**

(0, X'000') Nenhuma razão a relatar.

Se o código de conclusão for MQCC\_FAILED ou MQCC\_WARNING, a função de saída poderá configurar o campo de código de razão como qualquer valor MQRC\_\* válido.

## **Chamada de linguagem C**

O gerenciador de filas define logicamente as variáveis a seguir:

```
MQAXP      ExitParms;      /* Exit parameter structure */
MQAXC      ExitContext;   /* Exit context structure */
MQHCONN    Hconn;        /* Connection handle */
PMQOD      pObjDesc;     /* Ptr to object descriptor */
PMQMD      pMsgDesc;     /* Ptr to message descriptor */
PMQPMO     pPutMsgOpts;  /* Ptr to put message options */
MQLONG     BufferLength;  /* Message buffer length */
PMQBYTE    pBuffer;      /* Ptr to message data */
MQLONG     CompCode;     /* Completion code */
MQLONG     Reason;       /* Reason code */
```

O gerenciador de filas chama logicamente a saída da seguinte forma:

```
MQ_PUT1_EXIT (&ExitParms, &ExitContext, &Hconn, &pObjDesc, &pMsgDesc,
              &pPutMsgOpts, &BufferLength, &pBuffer, &CompCode, &Reason)
```

Sua saída deve corresponder ao seguinte protótipo de função C:

```
void MQENTRY MQ_PUT1_EXIT (
PMQAXP      pExitParms,   /* Address of exit parameter structure */
PMQAXC      pExitContext, /* Address of exit context structure */
PMQHCONN    pHconn,      /* Address of connection handle */
PPMQOD      ppObjDesc,   /* Address of ptr to object descriptor */
PPMQMD      ppMsgDesc,   /* Address of ptr to message descriptor */
PPMQPMO     ppPutMsgOpts, /* Address of ptr to put message options */
PMQLONG     pBufferLength, /* Address of message buffer length */
PPMQBYTE    ppBuffer,    /* Address of ptr to message buffer */
PMQLONG     pCompCode,   /* Address of completion code */
PMQLONG     pReason);    /* Address of reason code qualifying
                           completion code */
```

### **Conjunto-MQ\_SET\_EXIT**

MQ\_SET\_EXIT fornece uma função de saída de conjunto para executar *antes* e *após* o processamento de chamada MQSET. Use o identificador de função MQXF\_SET com as razões de saída MQXR\_BEFORE e MQXR\_AFTER para registrar *antes* e *após* funções de saída de chamada MQSET.

A interface para esta função é:

```
MQ_SET_EXIT (&ExitParms, &ExitContext, &Hconn, &Hobj, &SelectorCount,  
            &pSelectors, &IntAttrCount, &pIntAttrs, &CharAttrLength,  
            &pCharAttr, &CompCode, &Reason)
```

em que os parâmetros são:

**ExitParms (MQAXP)-entrada/saída**

Estrutura do parâmetro de saída

**ExitContext (MQAXC)-entrada/saída**

Estrutura do contexto de saída

**Hconn (MQHCONN)-entrada**

Manipulação de conexões.

**Hobj (MQHOBJ)-entrada**

Manipulação de objetos.

**SelectorCount (MQLONG)-entrada**

Contagem de seletores.

**pSelectors (PMQLONG)-entrada / saída**

Ponteiro para a matriz de valores do seletor

**IntAttrContagem (MQLONG)-entrada**

Contagem de atributos de número inteiro.

**Attrs pInt(PMQLONG)-entrada/saída**

Ponteiro para matriz de valores de atributo de número inteiro.

**CharAttrComprimento (MQLONG)-entrada/saída.**

Comprimento da matriz de atributos de caracteres

**pCharAttrs (PMQCHAR)-entrada/saída**

Ponteiro para valores de atributo de caractere.

**CompCode (MQLONG)-entrada/saída.**

Código de conclusão, valores válidos para os quais são:

**MQCC\_OK**

Indica conclusão bem-sucedida.

**MQCC\_WARNING**

Conclusão parcial.

**MQCC\_FAILED**

Chamada com falha

**Razão (MQLONG)-entrada/saída**

Código de razão qualificando o código de conclusão.

Se o código de conclusão for MQCC\_OK, o único valor válido será:

**MQRC\_NONE**

(0, X'000') Nenhuma razão a relatar.

Se o código de conclusão for MQCC\_FAILED ou MQCC\_WARNING, a função de saída poderá configurar o campo de código de razão como qualquer valor MQRC\_\* válido.

## Chamada de linguagem C

O gerenciador de filas define logicamente as variáveis a seguir:

```
MQAXP      ExitParms;      /* Exit parameter structure */  
MQAXC      ExitContext;    /* Exit context structure */  
MQHCONN    Hconn;         /* Connection handle */  
MQHOBJ     Hobj;          /* Object handle */  
MQLONG     SelectorCount; /* Count of selectors */  
PMQLONG    pSelectors;    /* Ptr to array of attribute selectors */  
MQLONG     IntAttrCount;  /* Count of integer attributes */
```

```

PMQLONG  pIntAttrs;          /* Ptr to array of integer attributes */
MQLONG   CharAttrLength;    /* Length of char attributes array */
PMQCHAR  pCharAttrs;        /* Ptr to character attributes */
MQLONG   CompCode;          /* Completion code */
MQLONG   Reason;            /* Reason code qualifying completion code */

```

O gerenciador de filas chama logicamente a saída da seguinte forma:

```

MQ_SET_EXIT (&ExitParms, &ExitContext, &Hconn, &Hobj, &SelectorCount,
             &pSelectors, &IntAttrCount, &pIntAttrs, &CharAttrLength,
             &pCharAttrs, &CompCode, &Reason)

```

Sua saída deve corresponder ao seguinte protótipo de função C:

```

void MQENTRY MQ_SET_EXIT (
PMQAXP   pExitParms,        /* Address of exit parameter structure */
PMQAXC   pExitContext,     /* Address of exit context structure */
PMQHCONN pHconn,           /* Address of connection handle */
PMQHOBJ  pHobj,            /* Address of object handle */
PMQLONG  pSelectorCount,   /* Address of selector count */
PPMQLONG ppSelectors,      /* Address of ptr to array of selectors */
PMQLONG  pIntAttrCount;    /* Address of count of integer attributes */
PPMQLONG ppIntAttrs,       /* Address of ptr to array of integer attributes */
PMQLONG  pCharAttrLength,  /* Address of character attribute length */
PPMQCHAR ppCharAttrs,     /* Address of ptr to character attributes array */
PMQLONG  pCompCode,        /* Address of completion code */
PMQLONG  pReason;          /* Address of reason code qualifying completion
                           code */

```

### Status-MQ\_STAT\_EXIT

MQ\_STAT\_EXIT fornece uma função de saída de status para executar *antes* e *após* processamento de chamada MQSTAT. Use o identificador de função MQXF\_STAT com as razões de saída MQXR\_BEFORE e MQXR\_AFTER para registrar *antes* e *após* funções de saída de chamada MQSTAT.

A interface para esta função é:

```

MQ_STAT_EXIT (&ExitParms, &ExitContext, &Hconn, &Type, &pStatus,
             &CompCode, &Reason)

```

em que os parâmetros são:

#### ExitParms (MQAXP)-entrada/saída

Estrutura do parâmetro de saída

#### ExitContext (MQAXC)-entrada/saída

Estrutura do contexto de saída

#### Hconn (MQHCONN)-entrada

Manipulação de conexões.

#### Tipo (MQLONG)-entrada

Tipo de informações de status para recuperar.

#### pStatus (PMQSTS)-saída

Ponteiro para o buffer de status

#### CompCode (MQLONG)-entrada/saída.

Código de conclusão, valores válidos para os quais são:

##### MQCC\_OK

Indica conclusão bem-sucedida.

##### MQCC\_WARNING

Conclusão parcial.

##### MQCC\_FAILED

Chamada com falha

### **Razão (MQLONG)-entrada/saída**

Código de razão qualificando o código de conclusão.

Se o código de conclusão for MQCC\_OK, o único valor válido será:

#### **MQRC\_NONE**

(0, X'000') Nenhuma razão a relatar.

Se o código de conclusão for MQCC\_FAILED ou MQCC\_WARNING, a função de saída poderá configurar o campo de código de razão como qualquer valor MQRC\_\* válido.

## **Chamada de linguagem C**

Sua saída deve corresponder ao seguinte protótipo de função C:

```
void MQENTRY MQ_STAT_EXIT (
PMQAXP    pExitParms,      /* Address of exit parameter structure */
PMQAXC    pExitContext,    /* Address of exit context structure */
PMQHCONN  pHconn,         /* Address of connection handle */
PMQLONG   pType,           /* Address of status type */
PPMQSTS   ppStatus,       /* Address of status buffer */
PMQLONG   pCompCode,      /* Address of completion code */
PMQLONG   pReason);       /* Address of reason code qualifying completion
                           code */
```

### **Finalização-MQ\_TERM\_EXIT**

MQ\_TERM\_EXIT fornece a finalização do nível de conexão, registrada com um identificador de função MQXF\_TERM e ExitReason MQXR\_CONNECTION. Se estiver registrado, MQ\_TERM\_EXIT será chamado uma vez para cada solicitação de desconexão

Como parte da finalização, o armazenamento não mais necessário pela saída pode ser liberado e qualquer limpeza necessária pode ser executada.

Se um MQ\_TERM\_EXIT falhar com MQXCC\_FAILED, o gerenciador de filas retornará do MQDISC que o chamou com MQCC\_FAILED e MQRC\_API\_EXIT\_ERROR.

Se o gerenciador de filas encontrar um erro ao finalizar o ambiente de execução da função de saída de API após chamar o último MQ\_TERM\_EXIT, o gerenciador de filas retornará da chamada MQDISC que chamou MQ\_TERM\_EXIT com MQCC\_FAILED e MQRC\_API\_EXIT\_TERM\_ERROR

A interface para esta função é:

```
MQ_TERM_EXIT (&ExitParms, &ExitContext, &CompCode, &Reason)
```

em que os parâmetros são:

#### **ExitParms (MQAXP)-entrada/saída**

Estrutura do parâmetro de saída

#### **ExitContext (MQAXC)-entrada/saída**

Estrutura do contexto de saída

#### **CompCode (MQLONG)-entrada/saída.**

Código de conclusão, valores válidos para os quais são:

#### **MQCC\_OK**

Indica conclusão bem-sucedida.

#### **MQCC\_FAILED**

Chamada com falha

#### **Razão (MQLONG)-entrada/saída**

Código de razão qualificando o código de conclusão.

Se o código de conclusão for MQCC\_OK, o único valor válido será:

## **MQRC\_NONE**

(0, X'000') Nenhuma razão a relatar.

Se o código de conclusão for MQCC\_FAILED, a função de saída poderá configurar o campo de código de razão para qualquer valor MQRC\_\* válido.

O CompCode e Reason retornados para o aplicativo dependem do valor do campo ExitResponse em MQAXP.

## **Chamada de linguagem C**

O gerenciador de filas define logicamente as variáveis a seguir:

```
MQAXP      ExitParms;      /* Exit parameter structure */
MQAXC      ExitContext;    /* Exit context structure */
MQQLONG    CompCode;      /* Completion code */
MQQLONG    Reason;        /* Reason code */
```

O gerenciador de filas chama logicamente a saída da seguinte forma:

```
MQ_TERM_EXIT (&ExitParms, &ExitContext, &CompCode, &Reason)
```

Sua saída deve corresponder ao seguinte protótipo de função C:

```
void MQENTRY MQ_TERM_EXIT (
MQAXP      pExitParms,      /* Address of exit parameter structure */
MQAXC      pExitContext,    /* Address of exit context structure */
MQQLONG    pCompCode,      /* Address of completion code */
MQQLONG    pReason);       /* Address of reason code qualifying
                             completion code */
```

## **Observações de Uso**

1. A função MQ\_TERM\_EXIT é opcional. Não é necessário que um conjunto de saída registre uma saída de finalização se não houver nenhum processamento de finalização a ser feito

Se as funções pertencentes ao conjunto de saída adquirirem recursos durante a conexão, uma função MQ\_TERM\_EXIT será um ponto conveniente no qual liberar esses recursos, por exemplo, liberar armazenamento obtido dinamicamente.

2. Se uma função MQ\_TERM\_EXIT for registrada quando a chamada MQDISC for emitida, a função de saída será chamada após todas as funções de saída MQDISC terem sido chamadas.
3. Se MQ\_TERM\_EXIT retornar MQXCC\_FAILED no campo ExitResponse do MQAXP ou falhar de alguma outra maneira, a chamada MQDISC que fez com que MQ\_TERM\_EXIT fosse chamado também falhará, com os parâmetros **CompCode** e **Reason** configurados para os valores apropriados

## **Registrar assinatura-MQ\_SUB\_EXIT**

MQ\_SUB\_EXIT fornece uma função de saída para executar *antes* e *após* o processamento de novo registro de assinatura. Use o identificador de função MQXF\_SUB com as razões de saída MQXR\_BEFORE e MQXR\_AFTER para registrar *antes* e *depois* das funções de saída de chamada de registro de assinatura.

A interface para esta função é:

```
MQ_SUB_EXIT (&ExitParms, &ExitContext, &Hconn, &pSubDesc, &pHobj, &pHsub, &CompCode, &Reason)
```

em que os parâmetros são:

### **ExitParms (MQAXP)-entrada/saída**

Estrutura do parâmetro de saída

### **ExitContext (MQAXC)-entrada/saída**

Estrutura do contexto de saída

### **Hconn (MQHCONN)-entrada / saída**

Manipulação de conexões.

### **pSubDesc-entrada/saída**

Matriz de seletores de atributo..

### **pHobj -entrada/saída**

Manipulação de objetos

### **pHsub (MQHOBJ) entrada/saída**

Identificador de assinatura

### **CompCode (MQLONG)-entrada/saída.**

Código de conclusão, valores válidos para os quais são:

#### **MQCC\_OK**

Indica conclusão bem-sucedida.

#### **MQCC\_WARNING**

Conclusão parcial.

#### **MQCC\_FAILED**

Chamada com falha

### **Razão (MQLONG)-entrada/saída**

Código de razão qualificando o código de conclusão.

Se o código de conclusão for MQCC\_OK, o único valor válido será:

#### **MQRC\_NONE**

(0, X'000') Nenhuma razão a relatar.

Se o código de conclusão for MQCC\_FAILED ou MQCC\_WARNING, a função de saída poderá configurar o campo de código de razão como qualquer valor MQRC\_\* válido.

## **Chamada de linguagem C**

O gerenciador de filas define logicamente as variáveis a seguir:

```
MQAXP      ExitParms;      /* Exit parameter structure */
MQAXC      ExitContext;    /* Exit context structure */
MQHCONN    Hconn;         /* Connection handle */
PMQSD      pSubDesc;      /* Subscription descriptor */
PMQHOBJS   pHobj;        /* Object Handle */
PMQHOBJS   pHsub;        /* Subscription handle */
MQLONG     CompCode;      /* Completion code */
MQLONG     Reason;       /* Reason code qualifying completion code */
```

O gerenciador de filas chama logicamente a saída da seguinte forma:

```
MQ_SUB_EXIT (&ExitParms, &ExitContext, &Hconn, &pSubDesc, &pHobj, &pHsub,
             &CompCode, &Reason);
```

Sua saída deve corresponder ao seguinte protótipo de função C:

```
PMQAXP     pExitParms;     /* Exit parameter structure */
PMQAXC     pExitContext;   /* Exit context structure */
PMQHCONN   pHconn;        /* Connection handle */
PPMQSD     ppSubDesc;     /* Subscription descriptor */
PPMQHOBJS  ppHobj;        /* Object Handle */
PPMQHOBJS  ppHsub;        /* Subscription handle */
PMQLONG    pCompCode;     /* Completion code */
PMQLONG    pReason;       /* Reason code qualifying completion code */
```

### **Solicitação de assinatura-MQ\_SUBRQ\_EXIT**

MQ\_SUBRQ\_EXIT fornece uma função de saída de solicitação de assinatura para executar *antes* e *após* o processamento de solicitação de assinatura. Use o identificador de função MQXF\_SUBRQ com as razões



de saída MQXR\_BEFORE e MQXR\_AFTER para registrar *antes* e *depois* das funções de saída de chamada de pedido de assinatura..

A interface para esta função é:

```
MQ_SUBRQ_EXIT (&ExitParms, &ExitContext, &Hconn, &pHsub, &Action, &pSubRqOpts,  
              &CompCode, &Reason)
```

em que os parâmetros são:

**ExitParms (MQAXP)-entrada/saída**

Estrutura do parâmetro de saída

**ExitContext (MQAXC)-entrada/saída**

Estrutura do contexto de saída

**Hconn (MQHCONN)-entrada / saída**

Manipulação de conexões.

**pHsub (MQHOBJ) entrada/saída**

Identificador de assinatura

**Entrada / saída da ação (MQLONG).**

Ação

**pSubRqOpts (MQSRO) entrada/saída**

**CompCode (MQLONG)-entrada/saída.**

Código de conclusão, valores válidos para os quais são:

**MQCC\_OK**

Indica conclusão bem-sucedida.

**MQCC\_WARNING**

Conclusão parcial.

**MQCC\_FAILED**

Chamada com falha

**Razão (MQLONG)-entrada/saída**

Código de razão qualificando o código de conclusão.

Se o código de conclusão for MQCC\_OK, o único valor válido será:

**MQRC\_NONE**

(0, X'000') Nenhuma razão a relatar.

Se o código de conclusão for MQCC\_FAILED ou MQCC\_WARNING, a função de saída poderá configurar o campo de código de razão como qualquer valor MQRC\_\* válido.

## Chamada de linguagem C

O gerenciador de filas define logicamente as variáveis a seguir:

```
MQAXP      ExitParms;      /* Exit parameter structure */  
MQAXC      ExitContext;    /* Exit context structure */  
MQHCONN     Hconn;         /* Connection handle */  
PMQLONG     pHsub;         /* Subscription handle */  
MQLONG      Action;        /* Action */  
PMQSRO      pSubRqOpts;    /* Subscription Request Options */  
MQLONG      CompCode;      /* Completion code */  
MQLONG      Reason;        /* Reason code qualifying completion code */
```

O gerenciador de filas chama logicamente a saída da seguinte forma:

```
MQ_SUBRQ_EXIT (&ExitParms, &ExitContext, &Hconn, &pHsub, &Action, &pSubRqOpts,  
              &CompCode, &Reason);
```

Sua saída deve corresponder ao seguinte protótipo de função C:

```

void MQENTRY MQ_SUBRQ_EXIT (
PMQAXP    pExitParms,      /* Address of exit parameter structure */
PMQAXC    pExitContext,   /* Address of exit context structure */
PMQHCONN  pHconn,         /* Address of connection handle */
PPMQHOBJ  ppHsub;         /* Address of Subscription handle */
PMQLONG   pAction;        /* Address of Action */
PPMQSRO   ppSubRqOpts;    /* Address of Subscription Request Options */
PMQLONG   pCompCode,      /* Address of completion code */
PMQLONG   pReason;        /* Address of reason code qualifying completion
                           code */

```

### ***xa\_close-XA\_CLOSE\_EXIT***

XA\_CLOSE\_EXIT fornece uma função de saída xa\_close para executar o processamento antes e depois de xa\_close. Use o identificador de função MQXF\_XACLOSE com as razões de saída MQXR\_BEFORE e MQXR\_AFTER para registrar as funções de saída de chamada xa\_close antes e depois...

A interface para esta função é:

```
XA_CLOSE_EXIT (&ExitParms, &ExitContext, &Hconn, &pXa_info, &Rmid, &Flags, &XARetCode)
```

em que os parâmetros são:

#### **ExitParms (MQAXP)-entrada/saída**

Estrutura do parâmetro de saída

#### **ExitContext (MQAXC)-entrada/saída**

Estrutura do contexto de saída

#### **Hconn (MQHCONN)-entrada**

Manipulação de conexões.

#### **pXa\_info (PMQCHAR)-entrada / saída**

Informações do gerenciador de recursos específico da instância

#### **Rmid (MQLONG)-entrada/saída**

Identificador do gerenciador de recursos.

#### **Sinalizadores (MQLONG)-entrada/saída.**

Opções do gerenciador de recursos

#### **XARetCode (MQLONG)-entrada/saída.**

Resposta da chamada XA.

## **Chamada de linguagem C**

O gerenciador de filas define logicamente as variáveis a seguir:

```

MQAXP    ExitParms;      /* Exit parameter structure */
MQAXC    ExitContext;   /* Exit context structure */
MQHCONN  Hconn;         /* Connection handle */
PMQCHAR  pXa_info;      /* Instance-specific RM info */
MQLONG   Rmid;          /* Resource manager identifier */
MQLONG   Flags;         /* Resource manager options*/
MQLONG   XARetCode;     /* Response from XA call */

```

O gerenciador de filas chama logicamente a saída da seguinte forma:

```
XA_CLOSE_EXIT (&ExitParms, &ExitContext, &Hconn, &pXa_info, &Rmid, &Flags, &XARetCode);
```

Sua saída deve corresponder ao seguinte protótipo de função C:

```

typedef void MQENTRY XA_CLOSE_EXIT (
PMQAXP    pExitParms,      /* Address of exit parameter structure */
PMQAXC    pExitContext,   /* Address of exit context structure */
PMQHCONN  pHconn,         /* Address of connection handle */
PPMQCHAR  ppXa_info,      /* Address of instance-specific RM info */

```

```

PMQLONG pRmid, /* Address of resource manager identifier */
PMQLONG pFlags, /* Address of resource manager options*/
PMQLONG pXARetCode); /* Address of response from XA call */

```

### ***xa\_commit-XA\_COMMIT\_EXIT***

XA\_COMMIT\_EXIT fornece uma função de saída xa\_commit para executar antes e depois do processamento xa\_commit. Use o identificador de função MQXF\_XACOMMIT com motivos de saída MQXR\_BEFORE e MQXR\_AFTER para registrar as funções de saída de chamada xa\_commit antes e depois.

A interface para esta função é:

```
XA_COMMIT_EXIT (&ExitParms, &ExitContext, &Hconn, &pXID, &Rmid, &Flags, &XARetCode)
```

em que os parâmetros são:

#### **ExitParms (MQAXP)-entrada/saída**

Estrutura do parâmetro de saída

#### **ExitContext (MQAXC)-entrada/saída**

Estrutura do contexto de saída

#### **Hconn (MQHCONN)-entrada**

Manipulação de conexões.

#### **pXID (MQPTR)-entrada/saída.**

ID da seção de transação.

#### **Rmid (MQLONG)-entrada/saída**

Identificador do gerenciador de recursos.

#### **Sinalizadores (MQLONG)-entrada/saída.**

Opções do gerenciador de recursos

#### **XARetCode (MQLONG)-entrada/saída.**

Resposta da chamada XA.

## **Chamada de linguagem C**

O gerenciador de filas define logicamente as variáveis a seguir:

```

MQAXP ExitParms; /* Exit parameter structure */
MQAXC ExitContext; /* Exit context structure */
MQHCONN Hconn; /* Connection handle */
MQPTR pXID; /* Transaction branch ID */
MQLONG Rmid; /* Resource manager identifier */
MQLONG Flags; /* Resource manager options*/
MQLONG XARetCode; /* Response from XA call */

```

O gerenciador de filas chama logicamente a saída da seguinte forma:

```
XA_COMMIT_EXIT (&ExitParms, &ExitContext, &Hconn, &pXID, &Rmid, &Flags, &XARetCode);
```

Sua saída deve corresponder ao seguinte protótipo de função C:

```

typedef void MQENTRY XA_COMMIT_EXIT (
PMQAXP pExitParms, /* Address of exit parameter structure */
PMQAXC pExitContext, /* Address of exit context structure */
PMQHCONN pHconn, /* Address of connection handle */
PMQPTR ppXID, /* Address of transaction branch ID */
PMQLONG pRmid, /* Address of resource manager identifier */
PMQLONG pFlags, /* Address of resource manager options*/
PMQLONG pXARetCode); /* Address of response from XA call */

```

## ***xa\_complete-XA\_COMPLETE\_EXIT***

XA\_COMPLETE\_EXIT fornece uma função de saída xa\_complete para executar o processamento antes e depois de xa\_complete. Use o identificador de função MQXF\_XACOMPLETE com as razões de saída MQXR\_BEFORE e MQXR\_AFTER para registrar as funções de saída de chamada xa\_complete.

A interface para esta função é:

```
XA_COMPLETE_EXIT (&ExitParms, &ExitContext, &Hconn, &pHandle, &pRetVal, &Rmid, &Flags, &XARetCode)
```

em que os parâmetros são:

### **ExitParms (MQAXP)-entrada/saída**

Estrutura do parâmetro de saída

### **ExitContext (MQAXC)-entrada/saída**

Estrutura do contexto de saída

### **Hconn (MQHCONN)-entrada**

Manipulação de conexões.

### **pHandle (PMQLONG)-entrada/saída**

Ponteiro para operação assíncrona.

### **pRetVal (PMQLONG)-entrada/saída**

Valor de retorno da operação assíncrona.

### **Rmid (MQLONG)-entrada/saída**

Identificador do gerenciador de recursos.

### **Sinalizadores (MQLONG)-entrada/saída.**

Opções do gerenciador de recursos

### **XARetCode (MQLONG)-entrada/saída.**

Resposta da chamada XA.

## **Chamada de linguagem C**

O gerenciador de filas define logicamente as variáveis a seguir:

```
MQAXP  ExitParms;    /* Exit parameter structure */
MQAXC  ExitContext; /* Exit context structure */
MQHCONN Hconn;      /* Connection handle */
PMQLONG pHandle;    /* Ptr to asynchronous op */
PMQLONG pRetVal;    /* Return value of async op */
MQLONG  Rmid;       /* Resource manager identifier */
MQLONG  Flags;      /* Resource manager options*/
MQLONG  XARetCode; /* Response from XA call */
```

O gerenciador de filas chama logicamente a saída da seguinte forma:

```
XA_COMPLETE_EXIT (&ExitParms, &ExitContext, &Hconn, &pHandle, &pRetVal, &Rmid, &Flags, &XARetCode);
```

Sua saída deve corresponder ao seguinte protótipo de função C:

```
typedef void MQENTRY XA_COMPLETE_EXIT (
  PMQAXP  pExitParms, /* Address of exit parameter structure */
  PMQAXC  pExitContext, /* Address of exit context structure */
  PMQHCONN pHconn, /* Address of connection handle */
  PPMQLONG ppHandle, /* Address of ptr to asynchronous op */
  PPMQLONG ppRetVal, /* Address of return value of async op */
  PMQLONG  pRmid, /* Address of resource manager identifier */
  PMQLONG  pFlags, /* Address of resource manager options*/
  PMQLONG  pXARetCode); /* Address of response from XA call */
```

## ***xa\_end-XA\_END\_EXIT***

XA\_END\_EXIT fornece uma função de saída *xa\_end* para executar antes e após o processamento de *xa\_end*. Use o identificador de função MQXF\_XAEND com as razões de saída MQXR\_BEFORE e MQXR\_AFTER para registrar as funções de saída de chamada *xa\_end* antes e após...

A interface para esta função é:

```
XA_END_EXIT (&ExitParms, &ExitContext, &Hconn, &pXID, &Rmid, &Flags, &XARetCode)
```

em que os parâmetros são:

### **ExitParms (MQAXP)-entrada/saída**

Estrutura do parâmetro de saída

### **ExitContext (MQAXC)-entrada/saída**

Estrutura do contexto de saída

### **Hconn (MQHCONN)-entrada**

Manipulação de conexões.

### **pXID (MQPTR)-entrada/saída.**

ID da seção de transação.

### **Rmid (MQLONG)-entrada/saída**

Identificador do gerenciador de recursos.

### **Sinalizadores (MQLONG)-entrada/saída.**

Opções do gerenciador de recursos

### **XARetCode (MQLONG)-entrada/saída.**

Resposta da chamada XA.

## **Chamada de linguagem C**

O gerenciador de filas define logicamente as variáveis a seguir:

```
MQAXP  ExitParms;    /* Exit parameter structure */
MQAXC  ExitContext; /* Exit context structure */
MQHCONN Hconn;      /* Connection handle */
MQPTR  pXID;        /* Transaction branch ID */
MQLONG Rmid;        /* Resource manager identifier */
MQLONG Flags;       /* Resource manager options*/
MQLONG XARetCode;  /* Response from XA call */
```

O gerenciador de filas chama logicamente a saída da seguinte forma:

```
XA_END_EXIT (&ExitParms, &ExitContext, &Hconn, &pXID, &Rmid, &Flags, &XARetCode);
```

Sua saída deve corresponder ao seguinte protótipo de função C:

```
typedef void MQENTRY XA_END_EXIT (
    PMQAXP  pExitParms, /* Address of exit parameter structure */
    PMQAXC  pExitContext, /* Address of exit context structure */
    PMQHCONN pHconn, /* Address of connection handle */
    PMQPTR  ppXID, /* Address of transaction branch ID */
    PMQLONG pRmid, /* Address of resource manager identifier */
    PMQLONG pFlags, /* Address of resource manager options*/
    PMQLONG pXARetCode); /* Address of response from XA call */
```

## ***xa\_forget-XA\_FORGET\_EXIT***

XA\_FORGET\_EXIT fornece uma função de saída *xa\_forget* para executar antes e depois do processamento de *xa\_forget*. Use o identificador de função MQXF\_XAFORGET com as razões de saída MQXR\_BEFORE e MQXR\_AFTER para registrar as funções de saída de chamada *xa\_forget*.

A interface para esta função é:

```
XA_FORGET_EXIT (&ExitParms, &ExitContext, &Hconn, &pXID, &Rmid, &Flags, &XARetCode)
```

em que os parâmetros são:

**ExitParms (MQAXP)-entrada/saída**

Estrutura do parâmetro de saída

**ExitContext (MQAXC)-entrada/saída**

Estrutura do contexto de saída

**Hconn (MQHCONN)-entrada**

Manipulação de conexões.

**pXID (MQPTR)-entrada/saída.**

ID da seção de transação.

**Rmid (MQLONG)-entrada/saída**

Identificador do gerenciador de recursos.

**Sinalizadores (MQLONG)-entrada/saída.**

Opções do gerenciador de recursos

**XARetCode (MQLONG)-entrada/saída.**

Resposta da chamada XA.

## Chamada de linguagem C

O gerenciador de filas define logicamente as variáveis a seguir:

```
MQAXP  ExitParms;    /* Exit parameter structure */
MQAXC  ExitContext; /* Exit context structure */
MQHCONN Hconn;      /* Connection handle */
MQPTR  pXID;        /* Transaction branch ID */
MQLONG Rmid;        /* Resource manager identifier */
MQLONG Flags;       /* Resource manager options*/
MQLONG XARetCode;  /* Response from XA call */
```

O gerenciador de filas chama logicamente a saída da seguinte forma:

```
XA_FORGET_EXIT (&ExitParms, &ExitContext, &Hconn, &pXID, &Rmid, &Flags, &XARetCode);
```

Sua saída deve corresponder ao seguinte protótipo de função C:

```
typedef void MQENTRY XA_FORGET_EXIT (
    PMQAXP  pExitParms, /* Address of exit parameter structure */
    PMQAXC  pExitContext, /* Address of exit context structure */
    PMQHCONN pHconn, /* Address of connection handle */
    PMQPTR  ppXID, /* Address of transaction branch ID */
    PMQLONG pRmid, /* Address of resource manager identifier */
    PMQLONG pFlags, /* Address of resource manager options*/
    PMQLONG pXARetCode); /* Address of response from XA call */
```

### **xa\_open-XA\_OPEN\_EXIT**

XA\_OPEN\_EXIT fornece uma função de saída xa\_open para executar antes e depois do processamento xa\_open.. Use o identificador de função MQXF\_XAOPEN com motivos de saída MQXR\_BEFORE e MQXR\_AFTER para registrar as funções de saída de chamada xa\_open antes e depois.

A interface para esta função é:

```
XA_OPEN_EXIT (&ExitParms, &ExitContext, &Hconn, &pXa_info, &Rmid, &Flags, &XARetCode)
```

em que os parâmetros são:

**ExitParms (MQAXP)-entrada/saída**

Estrutura do parâmetro de saída

**ExitContext (MQAXC)-entrada/saída**

Estrutura do contexto de saída

**Hconn (MQHCONN)-entrada**

Manipulação de conexões.

**pXa\_info (PMQCHAR)-entrada / saída**

Informações do gerenciador de recursos específico da instância

**Rmid (MQLONG)-entrada/saída**

Identificador do gerenciador de recursos.

**Sinalizadores (MQLONG)-entrada/saída.**

Opções do gerenciador de recursos

**XARetCode (MQLONG)-entrada/saída.**

Resposta da chamada XA.

**Chamada de linguagem C**

O gerenciador de filas define logicamente as variáveis a seguir:

```

MQAXP  ExitParms;    /* Exit parameter structure */
MQAXC  ExitContext; /* Exit context structure */
MQHCONN Hconn;      /* Connection handle */
PMQCHAR pXa_info;   /* Instance-specific RM info */
MQLONG Rmid;        /* Resource manager identifier */
MQLONG Flags;       /* Resource manager options*/
MQLONG XARetCode;   /* Response from XA call */

```

O gerenciador de filas chama logicamente a saída da seguinte forma:

```
XA_OPEN_EXIT (&ExitParms, &ExitContext, &Hconn, &pXa_info, &Rmid, &Flags, &XARetCode);
```

Sua saída deve corresponder ao seguinte protótipo de função C:

```

typedef void MQENTRY XA_OPEN_EXIT (
    PMQAXP  pExitParms, /* Address of exit parameter structure */
    PMQAXC  pExitContext, /* Address of exit context structure */
    PMQHCONN pHconn, /* Address of connection handle */
    PPMQCHAR ppXa_info, /* Address of instance-specific RM info */
    PMQLONG pRmid, /* Address of resource manager identifier */
    PMQLONG pFlags, /* Address of resource manager options*/
    PMQLONG pXARetCode); /* Address of response from XA call */

```

**xa\_prepare-XA\_PREPARE\_EXIT**

XA\_PREPARE\_EXIT fornece uma função de saída xa\_prepare para executar o processamento antes e depois de xa\_prepare. Use o identificador de função MQXF\_XAPREPARE com os motivos de saída MQXR\_BEFORE e MQXR\_AFTER para registrar as funções de saída de chamada xa\_prepare antes e depois.

A interface para esta função é:

```
XA_PREPARE_EXIT (&ExitParms, &ExitContext, &Hconn, &pXID, &Rmid, &Flags, &XARetCode)
```

em que os parâmetros são:

**ExitParms (MQAXP)-entrada/saída**

Estrutura do parâmetro de saída

**ExitContext (MQAXC)-entrada/saída**

Estrutura do contexto de saída

**Hconn (MQHCONN)-entrada**

Manipulação de conexões.

**pXID (MQPTR)-entrada/saída.**

ID da seção de transação.

**Rmid (MQLONG)-entrada/saída**

Identificador do gerenciador de recursos.

**Sinalizadores (MQLONG)-entrada/saída.**

Opções do gerenciador de recursos

**XARetCode (MQLONG)-entrada/saída.**

Resposta da chamada XA.

**Chamada de linguagem C**

O gerenciador de filas define logicamente as variáveis a seguir:

```
MQAXP  ExitParms; /* Exit parameter structure */
MQAXC  ExitContext; /* Exit context structure */
MQHCONN Hconn; /* Connection handle */
MQPTR  pXID; /* Transaction branch ID */
MQLONG Rmid; /* Resource manager identifier */
MQLONG Flags; /* Resource manager options*/
MQLONG XARetCode; /* Response from XA call */
```

O gerenciador de filas chama logicamente a saída da seguinte forma:

```
XA_PREPARE_EXIT (&ExitParms, &ExitContext, &Hconn, &pXID, &Rmid, &Flags, &XARetCode);
```

Sua saída deve corresponder ao seguinte protótipo de função C:

```
typedef void MQENTRY XA_PREPARE_EXIT (
    PMQAXP  pExitParms, /* Address of exit parameter structure */
    PMQAXC  pExitContext, /* Address of exit context structure */
    PMQHCONN pHconn, /* Address of connection handle */
    PMQPTR  ppXID, /* Address of transaction branch ID */
    PMQLONG pRmid, /* Address of resource manager identifier */
    PMQLONG pFlags, /* Address of resource manager options*/
    PMQLONG pXARetCode); /* Address of response from XA call */
```

***xa\_recover-XA\_RECOVER\_EXIT***

XA\_RECOVER\_EXIT fornece uma função de saída *xa\_recover* para executar antes e depois do processamento *xa\_recover*. Use o identificador de função MQXF\_XARECOVER com motivos de saída MQXR\_BEFORE e MQXR\_AFTER para registrar as funções de saída de chamada *xa\_recover* antes e depois.

A interface para esta função é:

```
XA_RECOVER_EXIT (&ExitParms, &ExitContext, &Hconn, &pXID, &Count, &Rmid, &Flags, &XARetCode)
```

em que os parâmetros são:

**ExitParms (MQAXP)-entrada/saída**

Estrutura do parâmetro de saída

**ExitContext (MQAXC)-entrada/saída**

Estrutura do contexto de saída

**Hconn (MQHCONN)-entrada**

Manipulação de conexões.

**pXID (MQPTR)-entrada/saída.**

ID da seção de transação.

**Contagem (MQLONG)-entrada/saída**

Máximo de XIDs na matriz XID



**Rmid (MQLONG)-entrada/saída**

Identificador do gerenciador de recursos.

**Sinalizadores (MQLONG)-entrada/saída.**

Opções do gerenciador de recursos

**XARetCode (MQLONG)-entrada/saída.**

Resposta da chamada XA.

**Chamada de linguagem C**

O gerenciador de filas define logicamente as variáveis a seguir:

```

MQAXP  ExitParms; /* Exit parameter structure */
MQAXC  ExitContext; /* Exit context structure */
MQHCONN Hconn; /* Connection handle */
MQPTR  pXID; /* Transaction branch ID */
MQLONG Count; /* Max XIDs in XID array */
MQLONG Rmid; /* Resource manager identifier */
MQLONG Flags; /* Resource manager options*/
MQLONG XARetCode; /* Response from XA call */

```

O gerenciador de filas chama logicamente a saída da seguinte forma:

```
XA_RECOVER_EXIT (&ExitParms, &ExitContext, &Hconn, &pXID, &Count, &Rmid, &Flags, &XARetCode);
```

Sua saída deve corresponder ao seguinte protótipo de função C:

```

typedef void MQENTRY XA_RECOVER_EXIT (
  PMQAXP  pExitParms, /* Address of exit parameter structure */
  PMQAXC  pExitContext, /* Address of exit context structure */
  PMQHCONN pHconn, /* Address of connection handle */
  MQPTR  ppXID, /* Address of transaction branch ID */
  PMQLONG pCount, /* Address of max XIDs in XID array */
  PMQLONG pRmid, /* Address of resource manager identifier */
  PMQLONG pFlags, /* Address of resource manager options*/
  PMQLONG pXARetCode); /* Address of response from XA call */

```

***xa\_rollback-XA\_ROLLBACK\_EXIT***

XA\_ROLLBACK\_EXIT fornece uma função de saída xa\_rollback para executar antes e depois do processamento xa\_rollback. Use o identificador de função MQXF\_XAROLLBACK com as razões de saída MQXR\_BEFORE e MQXR\_AFTER para registrar as funções de saída de chamada xa\_rollback antes e depois.

A interface para esta função é:

```
XA_ROLLBACK_EXIT (&ExitParms, &ExitContext, &Hconn, &pXID, &Rmid, &Flags, &XARetCode)
```

em que os parâmetros são:

**ExitParms (MQAXP)-entrada/saída**

Estrutura do parâmetro de saída

**ExitContext (MQAXC)-entrada/saída**

Estrutura do contexto de saída

**Hconn (MQHCONN)-entrada**

Manipulação de conexões.

**pXID (MQPTR)-entrada/saída.**

ID da seção de transação.

**Rmid (MQLONG)-entrada/saída**

Identificador do gerenciador de recursos.

### **Sinalizadores (MQLONG)-entrada/saída.**

Opções do gerenciador de recursos

### **XARetCode (MQLONG)-entrada/saída.**

Resposta da chamada XA.

## **Chamada de linguagem C**

O gerenciador de filas define logicamente as variáveis a seguir:

```
MQAXP  ExitParms;    /* Exit parameter structure */
MQAXC  ExitContext; /* Exit context structure */
MQHCONN Hconn;      /* Connection handle */
MQPTR  pXID;        /* Transaction branch ID */
MQLONG Rmid;        /* Resource manager identifier */
MQLONG Flags;       /* Resource manager options*/
MQLONG XARetCode;   /* Response from XA call */
```

O gerenciador de filas chama logicamente a saída da seguinte forma:

```
XA_ROLLBACK_EXIT (&ExitParms, &ExitContext, &Hconn, &pXID, &Rmid, &Flags, &XARetCode);
```

Sua saída deve corresponder ao seguinte protótipo de função C:

```
typedef void MQENTRY XA_ROLLBACK_EXIT (
    PMQAXP  pExitParms, /* Address of exit parameter structure */
    PMQAXC  pExitContext, /* Address of exit context structure */
    PMQHCONN pHconn, /* Address of connection handle */
    PMQPTR  ppXID, /* Address of transaction branch ID */
    PMQLONG pRmid, /* Address of resource manager identifier */
    PMQLONG pFlags, /* Address of resource manager options*/
    PMQLONG pXARetCode); /* Address of response from XA call */
```

### ***xa\_start-XA\_START\_EXIT***

XA\_START\_EXIT fornece uma função de saída *xa\_start* para executar o processamento antes e depois de *xa\_start*. Use o identificador de função MQXF\_XASTART com as razões de saída MQXR\_BEFORE e MQXR\_AFTER para registrar as funções de saída de chamada *xa\_start* antes e após...

A interface para esta função é:

```
XA_START_EXIT (&ExitParms, &ExitContext, &Hconn, &pXID, &Rmid, &Flags, &XARetCode)
```

em que os parâmetros são:

#### **ExitParms (MQAXP)-entrada/saída**

Estrutura do parâmetro de saída

#### **ExitContext (MQAXC)-entrada/saída**

Estrutura do contexto de saída

#### **Hconn (MQHCONN)-entrada**

Manipulação de conexões.

#### **pXID (MQPTR)-entrada/saída.**

ID da seção de transação.

#### **Rmid (MQLONG)-entrada/saída**

Identificador do gerenciador de recursos.

#### **Sinalizadores (MQLONG)-entrada/saída.**

Opções do gerenciador de recursos

#### **XARetCode (MQLONG)-entrada/saída.**

Resposta da chamada XA.

## Chamada de linguagem C

O gerenciador de filas define logicamente as variáveis a seguir:

```
MQAXP  ExitParms;    /* Exit parameter structure */
MQAXC  ExitContext; /* Exit context structure */
MQHCONN Hconn;      /* Connection handle */
MQPTR  pXID;        /* Transaction branch ID */
MQLONG Rmid;        /* Resource manager identifier */
MQLONG Flags;       /* Resource manager options*/
MQLONG XARetCode;   /* Response from XA call */
```

O gerenciador de filas chama logicamente a saída da seguinte forma:

```
XA_START_EXIT (&ExitParms, &ExitContext, &Hconn, &pXID, &Rmid, &Flags, &XARetCode);
```

Sua saída deve corresponder ao seguinte protótipo de função C:

```
typedef void MQENTRY XA_START_EXIT (
    PMQAXP  pExitParms, /* Address of exit parameter structure */
    PMQAXC  pExitContext, /* Address of exit context structure */
    PMQHCONN pHconn, /* Address of connection handle */
    MQPTR  ppXID, /* Address of transaction branch ID */
    PMQLONG pRmid, /* Address of resource manager identifier */
    PMQLONG pFlags, /* Address of resource manager options*/
    PMQLONG pXARetCode); /* Address of response from XA call */
```

### ***ax\_reg-AX\_REG\_EXIT***

AX\_REG\_EXIT fornece uma função de saída ax\_reg para executar o processamento antes e depois de ax\_reg Use o identificador de função MQXF\_AXREG com as razões de saída MQXR\_BEFORE e MQXR\_AFTER para registrar as funções de saída de chamada ax\_reg antes e depois.

A interface para esta função é:

```
AX_REG_EXIT (&ExitParms, &ExitContext, &pXID, &Rmid, &Flags, &XARetCode)
```

em que os parâmetros são:

#### **ExitParms (MQAXP)-entrada/saída**

Estrutura do parâmetro de saída

#### **ExitContext (MQAXC)-entrada/saída**

Estrutura do contexto de saída

#### **Hconn (MQHCONN)-entrada**

Manipulação de conexões.

#### **pXID (MQPTR)-entrada/saída.**

ID da seção de transação.

#### **Rmid (MQLONG)-entrada/saída**

Identificador do gerenciador de recursos.

#### **Sinalizadores (MQLONG)-entrada/saída.**

Opções do gerenciador de recursos

#### **XARetCode (MQLONG)-entrada/saída.**

Resposta da chamada XA.

## Chamada de linguagem C

O gerenciador de filas define logicamente as variáveis a seguir:

```
MQAXP  ExitParms;    /* Exit parameter structure */
MQAXC  ExitContext; /* Exit context structure */
MQPTR  pXID;        /* Transaction branch ID */
```

```

MQLONG Rmid;          /* Resource manager identifier */
MQLONG Flags;        /* Resource manager options*/
MQLONG XARetCode;    /* Response from XA call */

```

O gerenciador de filas chama logicamente a saída da seguinte forma:

```
AX_REG_EXIT (&ExitParms, &ExitContext, &pXID, &Rmid, &Flags, &XARetCode);
```

Sua saída deve corresponder ao seguinte protótipo de função C:

```

typedef void MQENTRY AX_REG_EXIT (
    PMQAXP pExitParms, /* Address of exit parameter structure */
    PMQAXC pExitContext, /* Address of exit context structure */
    PMQPTR ppXID, /* Address of transaction branch ID */
    PMQLONG pRmid, /* Address of resource manager identifier */
    PMQLONG pFlags, /* Address of resource manager options*/
    PMQLONG pXARetCode); /* Address of response from XA call */

```

### **ax\_unreg-AX\_UNREG\_EXIT**

AX\_UNREG\_EXIT fornece uma função de saída ax\_unreg para executar antes e depois do processamento ax\_unreg. Use o identificador de função MQXF\_AXUNREG com motivos de saída MQXR\_BEFORE e MQXR\_AFTER para registrar as funções de saída de chamada ax\_unreg antes e depois.

A interface para esta função é:

```
AX_UNREG_EXIT (&ExitParms, &ExitContext, &Rmid, &Flags, &XARetCode);
```

em que os parâmetros são:

#### **ExitParms (MQAXP)-entrada/saída**

Estrutura do parâmetro de saída

#### **ExitContext (MQAXC)-entrada/saída**

Estrutura do contexto de saída

#### **Rmid (MQLONG)-entrada/saída**

Identificador do gerenciador de recursos.

#### **Sinalizadores (MQLONG)-entrada/saída.**

Opções do gerenciador de recursos

#### **XARetCode (MQLONG)-entrada/saída.**

Resposta da chamada XA.

### **Chamada de linguagem C**

O gerenciador de filas define logicamente as variáveis a seguir:

```

MQAXP ExitParms; /* Exit parameter structure */
MQAXC ExitContext; /* Exit context structure */
MQLONG Rmid; /* Resource manager identifier */
MQLONG Flags; /* Resource manager options*/
MQLONG XARetCode; /* Response from XA call */

```

O gerenciador de filas chama logicamente a saída da seguinte forma:

```
AX_UNREG_EXIT (&ExitParms, &ExitContext, &Rmid, &Flags, &XARetCode);
```

Sua saída deve corresponder ao seguinte protótipo de função C:

```

typedef void MQENTRY AX_UNREG_EXIT (
    PMQAXP pExitParms, /* Address of exit parameter structure */
    PMQAXC pExitContext, /* Address of exit context structure */
    PMQLONG pRmid, /* Address of resource manager identifier */

```

```

PMQLONG pFlags,      /* Address of resource manager options*/
PMQLONG pXARetCode); /* Address of response from XA call */

```

## Informações gerais sobre funções de saída de chamada

Este tópico fornece alguma orientação geral para ajudá-lo a planejar suas saídas, particularmente relacionadas à manipulação de erros e eventos inesperados

### Falha de saída

Se uma função de saída for finalizada de forma anormal após uma chamada MQGET destrutiva, fora do ponto de sincronização, mas antes que a mensagem tenha sido transmitida para o aplicativo, o manipulador de saída poderá se recuperar da falha e passar o controle para o aplicativo...

Neste caso, a mensagem pode ser perdida. É como o que acontece quando um aplicativo falha imediatamente após receber uma mensagem de uma fila.

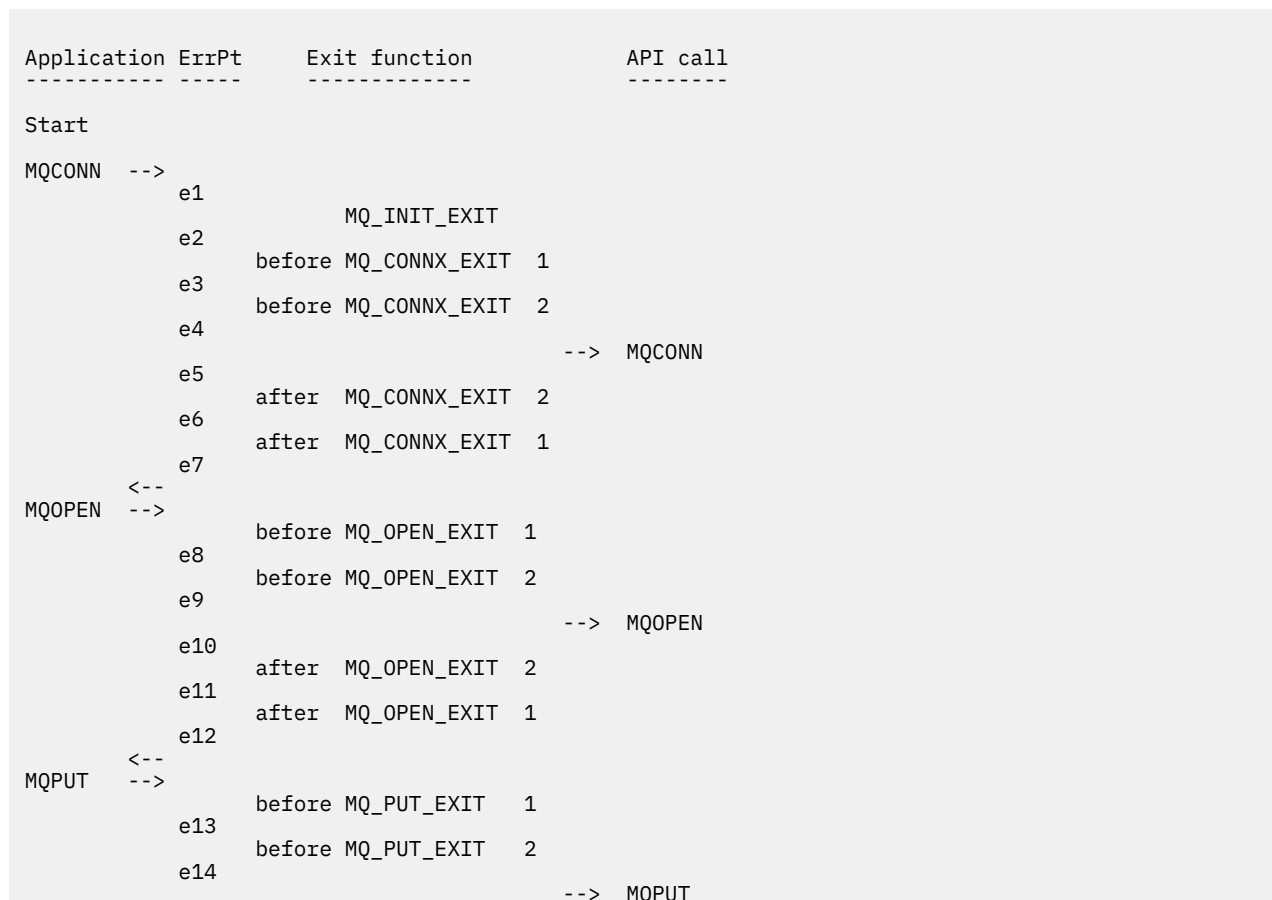
A chamada MQGET pode concluir com MQCC\_FAILED e MQRC\_API\_EXIT\_ERROR.

Se uma função de saída de chamada API *antes* for finalizada de forma anormal, o manipulador de saída poderá se recuperar da falha e passar o controle para o aplicativo sem processar a chamada API. Neste caso, a função de saída deve recuperar quaisquer recursos que ela possua.

Se as saídas encadeadas estiverem em uso, as saídas de chamada da API *após* para qualquer *antes* das saídas de chamada da API que foram acionadas com sucesso poderão ser acionadas. A chamada da API pode falhar com MQCC\_FAILED e MQRC\_API\_EXIT\_ERROR

### Exemplo de manipulação de erros para funções de saída

O diagrama a seguir mostra os pontos (e N) na qual erros podem ocorrer. Ele é apenas um exemplo para mostrar como as saídas se comportam e devem ser lidas junto com a tabela a seguir: Neste exemplo, duas funções de saída são chamadas antes e após cada chamada API para mostrar o comportamento com saídas encadeadas.



```

e15      after MQ_PUT_EXIT  2
e16      after MQ_PUT_EXIT  1
e17      <--
MQCLOSE  -->
e18      before MQ_CLOSE_EXIT 1
e19      before MQ_CLOSE_EXIT 2
e20      --> MQCLOSE
e21      after MQ_CLOSE_EXIT 2
e22      after MQ_CLOSE_EXIT 1
e23      <--
MQDISC  -->
e24      before MQ_DISC_EXIT 1
e25      before MQ_DISC_EXIT 2
e26      --> MQDISC
e27      after MQ_DISC_EXIT 2
e28      after MQ_DISC_EXIT 1
e29      <--
end

```

A tabela a seguir lista as ações a serem executadas em cada ponto de erro. Apenas um subconjunto dos pontos de erro foi coberto, pois as regras mostradas aqui podem se aplicar a todos os outros. São as ações que especificam o comportamento pretendido em cada caso.

<i>Tabela 837. Erros de saída de API e ações apropriadas para executar</i>		
<b>Err Pt</b>	<b>Descrição</b>	<b>Ações</b>
e1	Erro ao configurar a configuração do ambiente	<ol style="list-style-type: none"> <li>1. Desfazer configuração do ambiente conforme necessário</li> <li>2. Funções sem saída da unidade</li> <li>3. MQCONN com MQCC_FAILED, MQRC_API_EXIT_LOAD_ERROR</li> </ol>
e2	A função MQ_INIT_EXIT é concluída com: <ul style="list-style-type: none"> <li>• MQXCC_FAILED</li> <li>• MQXCC_*</li> </ul>	<ul style="list-style-type: none"> <li>• Para MQXCC_FAILED:               <ol style="list-style-type: none"> <li>1. Limpar o ambiente</li> <li>2. MQCONN com MQCC_FAILED, MQRC_API_EXIT_INIT_ERROR</li> </ol> </li> <li>• Para MQXCC_*               <ol style="list-style-type: none"> <li>1. Aja como para os valores de MQXCC_* e MQXR2_*<sup>1</sup></li> <li>2. Limpar o ambiente</li> </ol> </li> </ul>

Tabela 837. Erros de saída de API e ações apropriadas para executar (continuação)

Err Pt	Descrição	Ações
e3	<p>Antes A função MQ_CONNX_EXIT 1 é concluída com:</p> <ul style="list-style-type: none"> <li>• MQXCC_FAILED</li> <li>• MQXCC_*</li> </ul>	<ul style="list-style-type: none"> <li>• Para MQXCC_FAILED:               <ol style="list-style-type: none"> <li>1. função MQ_TERM_EXIT da unidade</li> <li>2. Limpar o ambiente</li> <li>3. Falha na chamada MQCONN com MQCC_FAILED, MQRC_API_EXIT_ERROR..</li> </ol> </li> <li>• Para MQXCC_*               <ol style="list-style-type: none"> <li>1. Aja como para os valores de MQXCC_* e MQXR2_*<sup>1</sup></li> <li>2. Função MQ_TERM_EXIT da unidade, se necessário</li> <li>3. Limpar o ambiente, se necessário</li> </ol> </li> </ul>
e4	<p>Antes A função MQ_CONNX_EXIT 2 é concluída com:</p> <ul style="list-style-type: none"> <li>• MQXCC_FAILED</li> <li>• MQXCC_*</li> </ul>	<ul style="list-style-type: none"> <li>• Para MQXCC_FAILED:               <ol style="list-style-type: none"> <li>1. Unidade <i>após</i> a função MQ_CONNX_EXIT 1</li> <li>2. função MQ_TERM_EXIT da unidade</li> <li>3. Limpar o ambiente</li> <li>4. Falha na chamada MQCONN com MQCC_FAILED, MQRC_API_EXIT_ERROR..</li> </ol> </li> <li>• Para MQXCC_*               <ol style="list-style-type: none"> <li>1. Aja como para os valores de MQXCC_* e MQXR2_*<sup>1</sup></li> <li>2. Unidade <i>após</i> a função MQ_CONNX_EXIT 1 se a saída não for suprimida</li> <li>3. Função MQ_TERM_EXIT da unidade, se necessário</li> <li>4. Limpar o ambiente, se necessário</li> </ol> </li> </ul>
e5	<p>A chamada MQCONN falha..</p>	<ol style="list-style-type: none"> <li>1. Passar MQCONN CompCode e Motivo</li> <li>2. Unidade <i>após</i> a função MQ_CONNX_EXIT 2 se o <i>before</i> MQ_CONNX_EXIT 2 for bem-sucedido e a saída não for suprimida</li> <li>3. Unidade <i>após</i> a função MQ_CONNX_EXIT 1 se o <i>anterior</i> MQ_CONNX_EXIT 1 foi bem-sucedido e a saída não foi suprimida</li> <li>4. função MQ_TERM_EXIT da unidade</li> <li>5. Limpar o ambiente</li> </ol>
e6	<p>Após a função MQ_CONNX_EXIT 2 ser concluída com:</p> <ul style="list-style-type: none"> <li>• MQXCC_FAILED</li> <li>• MQXCC_*</li> </ul>	<ul style="list-style-type: none"> <li>• Para MQXCC_FAILED:               <ol style="list-style-type: none"> <li>1. Unidade <i>após</i> a função MQ_CONNX_EXIT 1</li> <li>2. Concluir chamada MQCONN com MQCC_FAILED, MQRC_API_EXIT_ERROR</li> </ol> </li> <li>• Para MQXCC_*               <ol style="list-style-type: none"> <li>1. Aja como para os valores de MQXCC_* e MQXR2_*<sup>1</sup></li> <li>2. Unidade <i>após</i> a função MQ_CONNX_EXIT 1, se necessário.</li> </ol> </li> </ul>

Tabela 837. Erros de saída de API e ações apropriadas para executar (continuação)

Err Pt	Descrição	Ações
e7	<p>Após a função MQ_CONNX_EXIT 1 é concluída com:</p> <ul style="list-style-type: none"> <li>• MQXCC_FAILED</li> <li>• MQXCC_*</li> </ul>	<ul style="list-style-type: none"> <li>• Para MQXCC_FAILED, conclua a chamada MQCONN com MQCC_FAILED, MQRC_API_EXIT_ERROR</li> <li>• Para MQXCC_*, aja como para os valores de MQXCC_* e MQXR2_*<sup>1</sup></li> </ul>
e8	<p>Antes A função MQ_OPEN_EXIT 1 é concluída com:</p> <ul style="list-style-type: none"> <li>• MQXCC_FAILED</li> <li>• MQXCC_*</li> </ul>	<ul style="list-style-type: none"> <li>• Para MQXCC_FAILED, conclua a chamada MQOPEN com MQCC_FAILED, MQRC_API_EXIT_ERROR</li> <li>• Para MQXCC_*, aja como para os valores de MQXCC_* e MQXR2_*<sup>1</sup></li> </ul>
e9	<p>Antes A função MQ_OPEN_EXIT 2 é concluída com:</p> <ul style="list-style-type: none"> <li>• MQXCC_FAILED</li> <li>• MQXCC_*</li> </ul>	<ul style="list-style-type: none"> <li>• Para MQXCC_FAILED:               <ol style="list-style-type: none"> <li>1. Unidade <i>após</i> a função MQ_OPEN_EXIT 1</li> <li>2. Chamada MQOPEN completa com MQCC_FAILED, MQRC_API_EXIT_ERROR</li> </ol> </li> <li>• Para MQXCC_*, aja como para os valores de MQXCC_* e MQXR2_*<sup>1</sup></li> </ul>
e10	<p>A chamada MQOPEN falha</p>	<ol style="list-style-type: none"> <li>1. Passar MQOPEN CompCode e Razão</li> <li>2. Unidade <i>após</i> a função MQ_OPEN_EXIT 2 se a saída não for suprimida</li> <li>3. Unidade <i>após</i> a função MQ_OPEN_EXIT 1 se a saída não for suprimida e se as saídas encadeadas não forem suprimidas</li> </ol>
e11	<p>Após a função MQ_OPEN_EXIT 2 é concluída com:</p> <ul style="list-style-type: none"> <li>• MQXCC_FAILED</li> <li>• MQXCC_*</li> </ul>	<ul style="list-style-type: none"> <li>• Para MQXCC_FAILED:               <ol style="list-style-type: none"> <li>1. Unidade <i>após</i> a função MQ_OPEN_EXIT 1</li> <li>2. Chamada MQOPEN completa com MQCC_FAILED, MQRC_API_EXIT_ERROR</li> </ol> </li> <li>• Para MQXCC_*               <ol style="list-style-type: none"> <li>1. Aja como para os valores de MQXCC_* e MQXR2_*<sup>1</sup></li> <li>2. Unidade <i>após</i> a função MQ_OPEN_EXIT 1 se a saída não for suprimida</li> </ol> </li> </ul>
e25	<p>Após a função MQ_DISC_EXIT 2 é concluída com:</p> <ul style="list-style-type: none"> <li>• MQXCC_FAILED</li> <li>• MQXCC_*</li> </ul>	<ul style="list-style-type: none"> <li>• Para MQXCC_FAILED:               <ol style="list-style-type: none"> <li>1. Unidade <i>após</i> função MQ_DISC_EXIT 1</li> <li>2. função MQ_TERM_EXIT da unidade</li> <li>3. Limpar ambiente de execução de saída</li> <li>4. Conclua a chamada MQDISC com MQCC_FAILED, MQRC_API_EXIT_ERROR</li> </ol> </li> <li>• Para MQXCC_*               <ol style="list-style-type: none"> <li>1. Aja como para os valores de MQXCC_* e MQXR2_*<sup>1</sup></li> <li>2. função MQ_TERM_EXIT da unidade</li> <li>3. Limpar ambiente de execução de saída</li> </ol> </li> </ul>

**Nota:**



1. Os valores de MQXCC\_\* e MQXR2\_\* e suas ações correspondentes são definidos em Como os gerenciadores de filas processam funções de saída.

### **Campos ExitResponse configurados incorretamente**

Este tópico fornece informações sobre o que ocorreria quando o campo ExitResponse fosse configurado para qualquer coisa, exceto os valores suportados

Se o campo ExitResponse for configurado para um valor diferente de um dos valores suportados as seguintes ações se aplicam:

- Para uma função de saída da API MQCONN ou MQDISC *anterior* :
  - O valor ExitResponse2 é ignorado.
  - Nenhuma outra função de saída *antes* na cadeia de saída (se houver) é chamada; a própria chamada API não é emitida.
  - Para quaisquer saídas *antes* que foram chamadas com sucesso, as saídas *após* são chamadas em ordem reversa
  - Se registrado, as funções de saída de finalização para aquelas *antes* das funções de saída MQCONN ou MQDISC na cadeia que foram chamadas com sucesso são orientadas a limpar após essas funções de saída
  - A chamada MQCONN ou MQDISC falha com MQRC\_API\_EXIT\_ERROR..
- Para uma função de saída da API *before* IBM MQ diferente de MQCONN ou MQDISC:
  - O valor ExitResponse2 é ignorado.
  - Nenhuma função de conversão de dados *antes* ou *depois* na cadeia de saída (se houver) será chamada.
  - Para quaisquer saídas *antes* que foram chamadas com sucesso, as saídas *após* são chamadas em ordem reversa
  - A própria chamada API IBM MQ não é emitida.
  - A chamada da API IBM MQ falha com MQRC\_API\_EXIT\_ERROR
- Para uma função de saída da API MQCONN ou MQDISC *após* :
  - O valor ExitResponse2 é ignorado.
  - As funções de saída restantes que foram chamadas com êxito antes da chamada de API são chamadas em ordem inversa
  - Se registrado, as funções de saída de finalização para aquelas *antes* ou *após* funções de saída MQCONN ou MQDISC na cadeia que foram chamadas com sucesso são orientadas a limpar após a saída
  - Um CompCode mais grave de MQCC\_WARNING e o CompCode retornado pela saída são retornados ao aplicativo.
  - Um Motivo de MQRC\_API\_EXIT\_ERROR é retornado ao aplicativo.
  - A chamada da API IBM MQ foi emitida com sucesso
- Para uma função de saída de chamada API *após* IBM MQ diferente de MQCONN ou MQDISC:
  - O valor ExitResponse2 é ignorado.
  - As funções de saída restantes que foram chamadas com êxito antes da chamada de API são chamadas em ordem inversa
  - Um CompCode mais grave de MQCC\_WARNING e o CompCode retornado pela saída são retornados ao aplicativo.
  - Um Motivo de MQRC\_API\_EXIT\_ERROR é retornado ao aplicativo.
  - A chamada da API IBM MQ foi emitida com sucesso
- Para a conversão de dados *antes* na função get exit:
  - O valor ExitResponse2 é ignorado.

- As funções de saída restantes que foram chamadas com êxito antes da chamada de API são chamadas em ordem inversa
- A mensagem não é convertida e a mensagem não convertida é retornada ao aplicativo.
- Um CompCode mais grave de MQCC\_WARNING e o CompCode retornado pela saída são retornados ao aplicativo.
- Um Motivo de MQRC\_API\_EXIT\_ERROR é retornado ao aplicativo.
- A chamada da API IBM MQ foi emitida com sucesso

**Nota:** Como o erro é com a saída, é melhor retornar MQRC\_API\_EXIT\_ERROR do que MQRC\_NOT\_CONVERTED.


Se uma função de saída configurar o campo ExitResponse2 para um valor diferente de um dos valores suportados, um valor de MQXR2\_DEFAULT\_CONTINUATION será assumido no lugar.

## Informações de referência da interface de serviços instaláveis


Esta coleção de tópicos fornece informações de referência para os serviços instaláveis

As funções e tipos de dados são listados em ordem alfabética no grupo para cada tipo de serviço.

### Conceitos relacionados

 [Serviços e componentes instaláveis para UNIX, Linux e Windows](#)

 [Serviços e componentes instaláveis para IBM i](#)

 [Informações de referência da interface de serviços instaláveis para IBM i](#)

### Tarefas relacionadas

[Estendendo as instalações do gerenciador de filas](#)

 [Configurando serviços instaláveis](#)

## Como as funções são mostradas

Como as funções de serviços instaláveis são documentados

Para cada função, há uma descrição, incluindo o identificador de função (para MQZEP)

Os *parâmetros* são mostrados na ordem em que devem ocorrer. Todos devem estar presentes.

Cada nome de parâmetro é seguido por seu tipo de dado. Esses são os tipos de dados elementares descritos no [“Tipos de dados elementares”](#) na página 237

A chamada de linguagem C também é fornecida, após a descrição dos parâmetros

## MQZ\_AUTHENTICATE\_USER-Autenticar usuário

Essa função é fornecida por um componente de serviço de autorização MQZAS\_VERSION\_5 e é chamado pelo gerenciador de filas para autenticar um usuário, ou para configurar campos de contexto de identidade. Ele é chamado quando o contexto do aplicativo de usuário do IBM MQ é estabelecido

O contexto de aplicativos é estabelecido durante as chamadas de conexão no ponto em que o contexto do usuário do aplicativo é inicializado e em cada ponto em que o contexto do usuário do aplicativo é alterado. Cada vez que uma chamada de conexão é feita, as informações de contexto do usuário do aplicativo são readquiridas no campo *IdentityContext*

O identificador de função para esta função (para MQZEP) é MQZID\_AUTHENTICATE.

### Sintaxe

MQZ\_AUTHENTICATE\_USER ( *QMgrName* , *SecurityParms* , *ApplicationContext* , *IdentityContext* , *CorrelationPtr* , *ComponentData* , *Continuação* , *CompCode* , *Motivo* )

## Parâmetros

### QMgrName

Tipo: MQCHAR48 - entrada

Nome do gerenciador de filas. O nome do gerenciador de fila chamando o componente. Esse nome é preenchido com espaços em branco até o comprimento total do parâmetro; o nome não é terminado por um caractere nulo.

O nome do gerenciador de filas é transmitido para o componente para obter informações; a interface de serviço de autorização não requer que o componente o use de nenhuma maneira definida

### SecurityParms

Tipo: MQCSP-entrada

Parâmetros de segurança.. Dados relacionados ao ID do usuário, senha e tipo de autenticação. Se o atributo AuthenticationType da estrutura MQCSP for especificada como MQCSP\_AUTH\_USER\_ID\_AND\_PWD, o ID do usuário e a senha serão comparados com os campos equivalentes no parâmetro IdentityContext (MQZIC) para determinar se eles correspondem. Para obter informações adicionais, consulte [“MQCSP-Parâmetros de segurança” na página 343](#).

Durante uma chamada MQI MQCONN, esse parâmetro contém valores nulos ou padrão.

### ApplicationContext

Tipo: MQZAC-entrada

Contexto do aplicativo Dados relacionados com a aplicação de chamada. Consulte [MQZAC-Contexto do aplicativo](#) para obter detalhes

Durante cada chamada MQI MQCONN ou MQCONNX, as informações de contexto do usuário na estrutura MQZAC são readquiridas.

### IdentityContext

Tipo: MQZIC-entrada/saída

Contexto de identidade.. Na entrada para a função autenticar usuário, isso identifica o contexto de identidade atual. A função autenticar usuário pode mudar isso, nesse ponto o gerenciador de filas adota o novo contexto de identidade. Consulte [MQZIC-contexto de identidade](#) para obter mais detalhes sobre a estrutura MQZIC.

### CorrelationPtr

Tipo: MQPTR-saída

Indicador de correlação. Especifica o endereço dos dados de correlação. Esse ponteiro é transmitido subsequentemente para outras chamadas do OAM....

### ComponentData

Tipo: MQBYTE x ComponentDataComprimento-entrada/saída

Dados do componente.. Esses dados são mantidos pelo gerenciador de filas em nome desse componente específico; quaisquer mudanças feitas nele por qualquer uma das funções fornecidas por esse componente são preservadas e apresentadas na próxima vez em que uma das funções desse componente for chamada de.

O comprimento desta área de dados é passado pelo gerenciador de fila no parâmetro de Comprimento ComponentData chamada MQZ\_INIT\_AUTHORITY

### Continuação

Tipo: MQLONG - saída

Sinalizador de continuação É possível especificar os seguintes valores:

#### **MQZCI\_DEFAULT**

Continuação dependente de outros componentes..

#### **MQZCI\_STOP**

Não continue com o próximo componente

## CompCode

Tipo: MQLONG - saída

Código de conclusão. Deve ser um dos valores a seguir:

### **MQCC\_OK**

Indica conclusão bem-sucedida.

### **MQCC\_FAILED**

A chamada falhou.

## Razão

Tipo: MQLONG - saída

Código de razão qualificando *CompCode*.

Se *CompCode* for MQCC\_OK:

### **MQRC\_NONE**

(0, X'000') Nenhuma razão a relatar.

Se *CompCode* for MQCC\_FAILED:

### **MQRC\_SERVICE\_ERROR**

(2289, X'8F1') Ocorreu um erro inesperado ao acessar o serviço.

Para obter mais informações sobre esses códigos de razão, consulte [Mensagens e códigos de razão](#).

## Chamada C

```
MQZ_AUTHENTICATE_USER (QMgrName, SecurityParms, ApplicationContext,  
                        IdentityContext, &CorrelationPtr, ComponentData,  
                        &Continuation, &CompCode, &Reason);
```

Declare os parâmetros transmitidos para o serviço da seguinte forma:

```
MQCHAR48  QMgrName;           /* Queue manager name */  
MQCSP     SecurityParms;      /* Security parameters */  
MQZAC     ApplicationContext; /* Application context */  
MQZIC     IdentityContext;    /* Identity context */  
MQPTR     CorrelationPtr;     /* Correlation pointer */  
MQBYTE    ComponentData[n];  /* Component data */  
MQLONG    Continuation;      /* Continuation indicator set by  
                             component */  
MQLONG    CompCode;          /* Completion code */  
MQLONG    Reason;            /* Reason code qualifying CompCode */
```

## MQZ\_CHECK\_AUTHORITY-Verificar autoridade

Essa função é fornecida por um componente de serviço de autorização MQZAS\_VERSION\_1 e é iniciada pelo gerenciador de filas para verificar se uma entidade tem autoridade para executar uma ação ou ações específicas em um objeto especificado.

O identificador de função para esta função (para MQZEP) é MQZID\_CHECK\_AUTHORITY.

## Sintaxe

```
MQZ_CHECK_AUTHORITY( QMgrName , EntityName , EntityType , ObjectName ,  
ObjectType , Authority , ComponentData , Continuation , CompCode , Reason )
```

## Parâmetros

### **QMgrName**

Tipo: MQCHAR48 - entrada

Nome do gerenciador de filas. O nome do gerenciador de fila chamando o componente. Esse nome é preenchido com espaços em branco até o comprimento total do parâmetro; o nome não é terminado por um caractere nulo.

O nome do gerenciador de filas é transmitido para o componente para obter informações; a interface de serviço de autorização não requer que o componente o use de nenhuma maneira definida

### **EntityName**

Tipo: MQCHAR12 - entrada

Nome da entidade.. O nome da entidade cuja autorização para o objeto deve ser verificada O comprimento máximo da sequência é de 12 caracteres; se ela for mais curta do que ela for preenchida à direita com espaços em branco. O nome não é terminado por um caractere nulo.

Não é essencial que esta entidade seja conhecida pelo serviço de segurança subjacente. Se não for conhecido, as autorizações do grupo especial **nobody** (ao qual todas as entidades são consideradas pertencentes) serão usadas para a verificação. Um nome em branco é válido e pode ser usado dessa maneira.

### **EntityType**

Tipo: MQLONG - entrada

Tipo de entidade.. O tipo de entidade especificado por EntityName. Deve ser um dos valores a seguir:

#### **MQZAET\_PRINCIPAL**

Principal.

#### **MQZAET\_GROUP**

### **ObjectName**

Tipo: MQCHAR48 - entrada

Nome do objeto. O nome do objeto para o qual o acesso é necessário O comprimento máximo da sequência é de 48 caracteres; se for mais curto do que isso, será preenchido à direita com espaços em branco. O nome não é terminado por um caractere nulo.

Se *ObjectType* for MQOT\_Q\_MGR, esse nome será o mesmo que *QMgrName*

### **ObjectType**

Tipo: MQLONG - entrada

Tipo de objeto. O tipo de entidade especificado por *ObjectName* Deve ser um dos valores a seguir:

#### **MQOT\_AUTH\_INFO**

Informações de autenticação..

#### **MQOT\_CHANNEL**

Canal.

#### **MQOT\_CLNTCONN\_CHANEXOL**

Canal de conexão do cliente.

#### **MQOT\_LISTENER**

Ouvinte.

#### **MQOT\_NAMELIST**

Lista de nomes

#### **MQOT\_PROCESS**

process definition.

#### **MQOT\_Q**

Fila.

#### **MQOT\_Q\_MGR**

Gerenciador de Filas

#### **MQOT\_SERVICE**

Serviço.

## **Autoridade**

Tipo: MQLONG - entrada

A autoridade a ser verificada. Se uma autorização estiver sendo verificada, este campo será igual à operação de autorização apropriada (constante MQZAO\_\*) Se mais de uma autorização estiver sendo verificada, ela será OR bit a bit das constantes MQZAO\_\* correspondentes.

As autorizações a seguir se aplicam ao uso das chamadas MQI:

### **MQZAO\_CONNECT**

Capacidade de usar a chamada MQCONN

### **MQZAO\_BROWSE**

Capacidade de usar a chamada MQGET com uma opção de procura

Isso permite que a opção MQGMO\_BROWSE\_FIRST, MQGMO\_BROWSE\_MSG\_UNDER\_CURSOR ou MQGMO\_BROWSE\_NEXT seja especificada na chamada MQGET..

### **MQZAO\_INPUT**

Principal. Capacidade de usar a chamada MQGET com uma opção de entrada

Isso permite que a opção MQOO\_INPUT\_SHARED, MQOO\_INPUT\_EXCLUSIVE ou MQOO\_INPUT\_AS\_Q\_DEF seja especificada na chamada MQOPEN.

### **MQZAO\_OUTPUT**

Capacidade de usar a chamada MQPUT

Isso permite que a opção MQOO\_OUTPUT seja especificada na chamada MQOPEN..

### **MQZAO\_INQUIRE**

Capacidade de usar a chamada MQINQ..

Isso permite que a opção MQOO\_INQUIRE seja especificada na chamada MQOPEN..

### **MQZAO\_SET**

Capacidade de usar a chamada MQSET

Isso permite que a opção MQOO\_SET seja especificada na chamada MQOPEN..

### **MQZAO\_PASS\_IDENTITY\_CONTEXT**

Capacidade de passar o contexto de identidade

Isso permite que a opção MQOO\_PASS\_IDENTITY\_CONTEXT seja especificada na chamada MQOPEN e que a opção MQPMO\_PASS\_IDENTITY\_CONTEXT seja especificada nas chamadas MQPUT e MQPUT1 .

### **MQZAO\_PASS\_ALL\_CONTEXT**

Capacidade de passar todo o contexto

Isso permite que a opção MQOO\_PASS\_ALL\_CONTEXT seja especificada na chamada MQOPEN e a opção MQPMO\_PASS\_ALL\_CONTEXT seja especificada nas chamadas MQPUT e MQPUT1 .

### **MQZAO\_SET\_IDENTITY\_CONTEXT**

Capacidade de configurar o contexto de identidade

Isso permite que a opção MQOO\_SET\_IDENTITY\_CONTEXT seja especificada na chamada MQOPEN e a opção MQPMO\_SET\_IDENTITY\_CONTEXT seja especificada nas chamadas MQPUT e MQPUT1 .

### **MQZAO\_SET\_ALL\_CONTEXT**

Capacidade de configurar todo o contexto

Isso permite que a opção MQOO\_SET\_ALL\_CONTEXT seja especificada na chamada MQOPEN e a opção MQPMO\_SET\_ALL\_CONTEXT seja especificada nas chamadas MQPUT e MQPUT1 .

### **MQZAO\_ALTERNATE\_USER\_AUTHORITY**

Capacidade de usar a autoridade de usuário alternativa

Isso permite que a opção MQOO\_ALTERNATE\_USER\_AUTHORITY seja especificada na chamada MQOPEN, e a opção MQPMO\_ALTERNATE\_USER\_AUTHORITY seja especificada na chamada MQPUT1 .

#### **MQZAO\_ALL\_MQI**

Todas as autorizações de MQI

Isso ativa todas as autorizações..

As autorizações a seguir se aplicam à administração de um gerenciador de filas:

#### **MQZAO\_CREATE**

Capacidade de criar objetos de um tipo especificado

#### **MQZAO\_DELETE**

Capacidade de excluir um objeto especificado..

#### **MQZAO\_DISPLAY**

Capacidade de exibir os atributos de um objeto especificado..

#### **MQZAO\_CHANGE**

A capacidade de alterar os atributos de um objeto especificado.

#### **MQZAO\_CLEAR**

Capacidade de excluir todas as mensagens de uma fila especificada.

#### **MQZAO\_AUTORIZAR**

Capacidade de autorizar outros usuários para um objeto especificado..

#### **MQZAO\_CONTROL**

Capacidade de iniciar ou parar um objeto de canal listener, de serviço ou não cliente e a capacidade de executar ping de um objeto de canal não cliente.

#### **MQZAO\_CONTROL\_EXTENDED**

Capacidade de reconfigurar um número de sequência ou resolver uma mensagem pendente em um objeto de canal não cliente.

#### **MQZAO\_ALL\_ADMIN**

Capacidade de configurar o contexto de identidade

Todas as autorizações de administração, além de MQZAO\_CREATE

As autorizações a seguir se aplicam ao uso do MQI e à administração de um gerenciador de filas:

#### **MQZAO\_ALL**

Todas as autorizações, além de MQZAO\_CREATE

#### **MQZAO\_NONE**

Sem autorizações.

### **ComponentData**

Tipo: MQBYTE x ComponentDataComprimento-entrada/saída

Dados do componente.. Esses dados são mantidos pelo gerenciador de filas em nome desse componente específico; quaisquer mudanças feitas nele por qualquer uma das funções fornecidas por esse componente são preservados e apresentados na próxima vez em que uma dessas funções do componente for chamada

O comprimento dessa área de dados é passado pelo gerenciador de filas no parâmetro

**ComponentDataLength** da chamada MQZ\_INIT\_AUTHORITY

### **Continuação**

Tipo: MQLONG - saída

Indicador de continuação configurado por componente.. Os valores a seguir podem ser especificados:

#### **MQZCI\_DEFAULT**

Continuação dependente do gerenciador de filas..

Para MQZ\_CHECK\_AUTHORITY, isso tem o mesmo efeito que MQZCI\_STOP

## **MQZCI\_CONTINUE**

Continue com o próximo componente

## **MQZCI\_STOP**

Não continue com o próximo componente

Se a chamada para um componente falhar (ou seja *CompCode* retorna MQCC\_FAILED) e o parâmetro *Continuation* for MQZCI\_DEFAULT ou MQZCI\_CONTINUE, o gerenciador de filas continuará a chamar outros componentes se houver algum.

Se a chamada for bem-sucedida (ou seja, *CompCode* retorna MQCC\_OK), nenhum outro componente será chamado, independentemente da configuração de *Continuação*.

Se a chamada falhar e o parâmetro *Continuation* for MQZCI\_STOP, nenhum outro componente será chamado e o erro será retornado ao gerenciador de fila. Os componentes não têm conhecimento de chamadas anteriores, portanto, o parâmetro *Continuation* é sempre configurado como MQZCI\_DEFAULT antes da chamada.

## **CompCode**

Tipo: MQLONG - saída

Código de conclusão. Deve ser um dos valores a seguir:

### **MQCC\_OK**

Indica conclusão bem-sucedida.

### **MQCC\_FAILED**

A chamada falhou.

## **Razão**

Tipo: MQLONG - saída

Código de razão qualificando *CompCode*.

Se *CompCode* for MQCC\_OK:

### **MQRC\_NONE**

(0, X'000') Nenhuma razão a relatar.

Se *CompCode* for MQCC\_FAILED:

### **MQRC\_NOT\_AUTHORIZED**

(2035, X'7F3') Não autorizado para acesso.

### **MQRC\_SERVICE\_ERROR**

(2289, X'8F1') Ocorreu um erro inesperado ao acessar o serviço.

### **MQRC\_SERVICE\_NOT\_AVAILABLE**

(2285, X'8ED') Serviço subjacente não disponível.

Para obter mais informações sobre esses códigos de razão, consulte [conclusão da API e códigos de razão](#).

## **Chamada C**

```
MQZ_CHECK_AUTHORITY (QMgrName, EntityName, EntityType, ObjectName,  
                    ObjectType, Authority, ComponentData,  
                    &Continuation, &CompCode, &Reason);
```

Os parâmetros transmitidos para o serviço são declarados da seguinte forma:

```
MQCHAR48 QMgrName;          /* Queue manager name */  
MQCHAR12 EntityName;       /* Entity name */  
MQLONG   EntityType;       /* Entity type */  
MQCHAR48 ObjectName;      /* Object name */  
MQLONG   ObjectType;      /* Object type */  
MQLONG   Authority;       /* Authority to be checked */  
MQBYTE   ComponentData[n]; /* Component data */  
MQLONG   Continuation;    /* Continuation indicator set by
```



```

MQLONG      CompCode;          component */
MQLONG      Reason;           /* Completion code */
                                     /* Reason code qualifying CompCode */

```

## MQZ\_CHECK\_AUTHORITY\_2 -Verificar autoridade (estendido)

Essa função é fornecida por um componente de serviço de autorização MQZAS\_VERSION\_2 e é iniciada pelo gerenciador de filas para verificar se uma entidade possui autoridade para executar uma ação ou ações específicas em um objeto especificado.

O identificador de função para esta função (para MQZEP) é MQZID\_CHECK\_AUTHORITY.

MQZ\_CHECK\_AUTHORITY\_2 é como MQZ\_CHECK\_AUTHORITY, mas com o parâmetro **EntityName** substituído pelo parâmetro **EntityData**.

### Sintaxe

```

MQZ_CHECK_AUTHORITY_2( QMgrName , EntityData , EntityType , ObjectName ,
ObjectType , Authority , ComponentData , Continuation , CompCode , Reason )

```

### Parâmetros

#### QMgrName

Tipo: MQCHAR48 - entrada

Nome do gerenciador de filas. O nome do gerenciador de fila chamando o componente. Esse nome é preenchido com espaços em branco até o comprimento total do parâmetro; o nome não é terminado por um caractere nulo.

O nome do gerenciador de filas é transmitido para o componente para obter informações; a interface de serviço de autorização não requer que o componente o use de nenhuma maneira definida

#### EntityData

Tipo: MQZED-entrada

Dados da entidade.. Dados relacionados à entidade com autorização para o objeto que será verificado. Consulte "[MQZED-Descrição de entidade](#)" na página 1737 para obter detalhes.

Não é essencial que esta entidade seja conhecida pelo serviço de segurança subjacente. Se não for conhecido, as autorizações do grupo especial **nobody** (ao qual todas as entidades são consideradas pertencentes) serão usadas para a verificação. Um nome em branco é válido e pode ser usado dessa maneira.

#### EntityType

Tipo: MQLONG - entrada

Tipo de entidade.. O tipo de entidade especificado por *EntityData* Deve ser um dos valores a seguir:

#### **MQZAET\_PRINCIPAL**

Principal.

#### **MQZAET\_GROUP**

#### ObjectName

Tipo: MQCHAR48 - entrada

Nome do objeto. O nome do objeto para o qual o acesso é necessário O comprimento máximo da sequência é de 48 caracteres; se for mais curto do que isso, será preenchido à direita com espaços em branco. O nome não é terminado por um caractere nulo.

Se *ObjectType* for MQOT\_Q\_MGR, esse nome será o mesmo que *QMgrName*

#### ObjectType

Tipo: MQLONG - entrada

Tipo de objeto. O tipo de entidade especificado por *ObjectName* Deve ser um dos valores a seguir:

**MQOT\_AUTH\_INFO**

Informações de autenticação..

**MQOT\_CHANNEL**

Canal.

**MQOT\_CLNTCONN\_CHANEXOL**

Canal de conexão do cliente.

**MQOT\_LISTENER**

Ouvinte.

**MQOT\_NAMELIST**

Lista de nomes

**MQOT\_PROCESS**

process definition.

**MQOT\_Q**

Fila.

**MQOT\_Q\_MGR**

Gerenciador de Filas

**MQOT\_SERVICE**

Serviço.

**MQOT\_TOPIC**

:NONE.

**Autoridade**

Tipo: MQLONG - entrada

A autoridade a ser verificada Se uma autorização estiver sendo verificada, este campo será igual à operação de autorização apropriada (constante MQZAO\_\*) Se mais de uma autorização estiver sendo verificada, ela será OR bit a bit das constantes MQZAO\_\* correspondentes.

As autorizações a seguir se aplicam ao uso das chamadas MQI:

**MQZAO\_CONNECT**

Capacidade de usar a chamada MQCONN

**MQZAO\_BROWSE**

Capacidade de usar a chamada MQGET com uma opção de procura

Isso permite que a opção MQGMO\_BROWSE\_FIRST, MQGMO\_BROWSE\_MSG\_UNDER\_CURSOR ou MQGMO\_BROWSE\_NEXT seja especificada na chamada MQGET..

**MQZAO\_INPUT**

Principal. Capacidade de usar a chamada MQGET com uma opção de entrada

Isso permite que a opção MQOO\_INPUT\_SHARED, MQOO\_INPUT\_EXCLUSIVE ou MQOO\_INPUT\_AS\_Q\_DEF seja especificada na chamada MQOPEN.

**MQZAO\_OUTPUT**

Capacidade de usar a chamada MQPUT

Isso permite que a opção MQOO\_OUTPUT seja especificada na chamada MQOPEN..

**MQZAO\_INQUIRE**

Capacidade de usar a chamada MQINQ..

Isso permite que a opção MQOO\_INQUIRE seja especificada na chamada MQOPEN..

**MQZAO\_SET**

Capacidade de usar a chamada MQSET

Isso permite que a opção MQOO\_SET seja especificada na chamada MQOPEN..

**MQZAO\_PASS\_IDENTITY\_CONTEXT**

Capacidade de passar o contexto de identidade

Isso permite que a opção MQOO\_PASS\_IDENTITY\_CONTEXT seja especificada na chamada MQOPEN e que a opção MQPMO\_PASS\_IDENTITY\_CONTEXT seja especificada nas chamadas MQPUT e MQPUT1 .

#### **MQZAO\_PASS\_ALL\_CONTEXT**

Capacidade de passar todo o contexto

Isso permite que a opção MQOO\_PASS\_ALL\_CONTEXT seja especificada na chamada MQOPEN e a opção MQPMO\_PASS\_ALL\_CONTEXT seja especificada nas chamadas MQPUT e MQPUT1 .

#### **MQZAO\_SET\_IDENTITY\_CONTEXT**

Capacidade de configurar o contexto de identidade

Isso permite que a opção MQOO\_SET\_IDENTITY\_CONTEXT seja especificada na chamada MQOPEN e a opção MQPMO\_SET\_IDENTITY\_CONTEXT seja especificada nas chamadas MQPUT e MQPUT1 .

#### **MQZAO\_SET\_ALL\_CONTEXT**

Capacidade de configurar todo o contexto

Isso permite que a opção MQOO\_SET\_ALL\_CONTEXT seja especificada na chamada MQOPEN e a opção MQPMO\_SET\_ALL\_CONTEXT seja especificada nas chamadas MQPUT e MQPUT1 .

#### **MQZAO\_ALTERNATE\_USER\_AUTHORITY**

Capacidade de usar a autoridade de usuário alternativa

Isso permite que a opção MQOO\_ALTERNATE\_USER\_AUTHORITY seja especificada na chamada MQOPEN, e a opção MQPMO\_ALTERNATE\_USER\_AUTHORITY seja especificada na chamada MQPUT1 .

#### **MQZAO\_ALL\_MQI**

Todas as autorizações de MQI

Isso ativa todas as autorizações..

As autorizações a seguir se aplicam à administração de um gerenciador de filas:

#### **MQZAO\_CREATE**

Capacidade de criar objetos de um tipo especificado

#### **MQZAO\_DELETE**

Capacidade de excluir um objeto especificado..

#### **MQZAO\_DISPLAY**

Capacidade de exibir os atributos de um objeto especificado..

#### **MQZAO\_CHANGE**

A capacidade de alterar os atributos de um objeto especificado.

#### **MQZAO\_CLEAR**

Capacidade de excluir todas as mensagens de uma fila especificada.

#### **MQZAO\_AUTORIZAR**

Capacidade de autorizar outros usuários para um objeto especificado..

#### **MQZAO\_CONTROL**

Capacidade de iniciar ou parar um objeto de canal listener, de serviço ou não cliente e a capacidade de executar ping de um objeto de canal não cliente.

#### **MQZAO\_CONTROL\_EXTENDED**

Capacidade de reconfigurar um número de sequência ou resolver uma mensagem pendente em um objeto de canal não cliente.

#### **MQZAO\_ALL\_ADMIN**

Capacidade de configurar o contexto de identidade

Todas as autorizações de administração, além de MQZAO\_CREATE

As autorizações a seguir se aplicam ao uso do MQI e à administração de um gerenciador de filas:

**MQZAO\_ALL**

Todas as autorizações, além de MQZAO\_CREATE

**MQZAO\_NONE**

Sem autorizações.

**ComponentData**

Tipo: MQBYTE x ComponentDataComprimento-entrada/saída

Dados do componente.. Esses dados são mantidos pelo gerenciador de filas em nome desse componente específico; quaisquer mudanças feitas nele por qualquer uma das funções fornecidas por esse componente são preservados e apresentados na próxima vez em que uma dessas funções do componente for chamada

O comprimento dessa área de dados é passado pelo gerenciador de filas no parâmetro **ComponentDataLength** da chamada MQZ\_INIT\_AUTHORITY

**Continuação**

Tipo: MQLONG - saída

Indicador de continuação configurado por componente.. Os valores a seguir podem ser especificados:

**MQZCI\_DEFAULT**

Continuação dependente do gerenciador de filas..

Para MQZ\_CHECK\_AUTHORITY, isso tem o mesmo efeito que MQZCI\_STOP

**MQZCI\_CONTINUE**

Continue com o próximo componente

**MQZCI\_STOP**

Não continue com o próximo componente

**CompCode**

Tipo: MQLONG - saída

Código de conclusão. Deve ser um dos valores a seguir:

**MQCC\_OK**

Indica conclusão bem-sucedida.

**MQCC\_FAILED**

A chamada falhou.

**Razão**

Tipo: MQLONG - saída

Código de razão qualificando *CompCode*.

Se *CompCode* for MQCC\_OK:

**MQRC\_NONE**

(0, X'000') Nenhuma razão a relatar.

Se *CompCode* for MQCC\_FAILED:

**MQRC\_NOT\_AUTHORIZED**

(2035, X'7F3') Não autorizado para acesso.

**MQRC\_SERVICE\_ERROR**

(2289, X'8F1') Ocorreu um erro inesperado ao acessar o serviço.

**MQRC\_SERVICE\_NOT\_AVAILABLE**

(2285, X'8ED') Serviço subjacente não disponível.

Para obter mais informações sobre esses códigos de razão, consulte [conclusão da API e códigos de razão](#).

## Chamada C

```
MQZ_CHECK_AUTHORITY_2 (QMgrName, &EntityData, EntityType,  
ObjectName, ObjectType, Authority, ComponentData,  
&Continuation, &CompCode, &Reason);
```

Os parâmetros transmitidos para o serviço são declarados da seguinte forma:

```
MQCHAR48 QMgrName;          /* Queue manager name */  
MQZED EntityData;          /* Entity data */  
MQLONG EntityType;        /* Entity type */  
MQCHAR48 ObjectName;      /* Object name */  
MQLONG ObjectType;        /* Object type */  
MQLONG Authority;         /* Authority to be checked */  
MQBYTE ComponentData[n]; /* Component data */  
MQLONG Continuation;      /* Continuation indicator set by  
                           component */  
MQLONG CompCode;          /* Completion code */  
MQLONG Reason;            /* Reason code qualifying CompCode */
```

### MQZ\_CHECK\_PRIVILEGED-Verifique se o usuário é privilegiado

Essa função é fornecida por um componente de serviço de autorização MQZAS\_VERSION\_6 e é chamada pelo gerenciador de filas para determinar se um usuário especificado é um usuário privilegiado.

O identificador de função para essa função (para MQZEP) é MQZID\_CHECK\_PRIVILEGED.

#### Sintaxe

```
MQZ_CHECK_PRIVILEGED( QMgrName , EntityData , EntityType , ComponentData ,  
Continuation , CompCode , Reason )
```

#### Parâmetros

##### QMgrName

Tipo: MQCHAR48 - entrada

Nome do gerenciador de filas. O nome do gerenciador de fila chamando o componente. Esse nome é preenchido com espaços em branco até o comprimento total do parâmetro; o nome não é terminado por um caractere nulo.

O nome do gerenciador de filas é transmitido para o componente para obter informações; a interface de serviço de autorização não requer que o componente o use de nenhuma maneira definida

##### EntityData

Tipo: MQZED-entrada

Dados da entidade.. Dados relacionados à entidade que será verificada. Para obter mais informações, consulte [“MQZED-Descrição de entidade”](#) na página 1737.

##### EntityType

Tipo: MQLONG - entrada

Tipo de entidade.. O tipo de entidade especificado por EntityData Deve ser um dos valores a seguir:

##### **MQZAET\_PRINCIPAL**

Principal.

##### **MQZAET\_GROUP**

##### ComponentData

Tipo: MQBYTEExComponentDataLength -entrada/saída.

Dados do componente.. Esses dados são mantidos pelo gerenciador de filas em nome desse componente específico; quaisquer mudanças feitas nele por qualquer uma das funções fornecidas

por esse componente são preservados e apresentados na próxima vez em que uma dessas funções do componente for chamada

O comprimento dessa área de dados é passado pelo gerenciador de filas no parâmetro **ComponentDataLength** da chamada MQZ\_INIT\_AUTHORITY

### **Continuação**

Tipo: MQLONG - saída

Indicador de continuação configurado por componente.. Os valores a seguir podem ser especificados:

#### **MQZCI\_DEFAULT**

Continuação dependente do gerenciador de filas..

Para MQZ\_CHECK\_AUTHORITY, isso tem o mesmo efeito que MQZCI\_STOP

#### **MQZCI\_CONTINUE**

Continue com o próximo componente

#### **MQZCI\_STOP**

Não continue com o próximo componente

Se a chamada para um componente falhar (ou seja, *CompCode* retorna MQCC\_FAILED) e o parâmetro *Continuation* for MQZCI\_DEFAULT ou MQZCI\_CONTINUE, o gerenciador de filas continuará a chamar outros componentes se houver algum.

Se a chamada for bem-sucedida (ou seja, *CompCode* retorna MQCC\_OK), nenhum outro componente será chamado, independentemente da configuração de *Continuação* .

Se a chamada falhar e o parâmetro *Continuation* for MQZCI\_STOP, nenhum outro componente será chamado e o erro será retornado ao gerenciador de fila. Os componentes não têm conhecimento de chamadas anteriores, portanto, o parâmetro *Continuation* é sempre configurado como MQZCI\_DEFAULT antes da chamada.

### **CompCode**

Tipo: MQLONG - saída

Código de conclusão. Deve ser um dos valores a seguir:

#### **MQCC\_OK**

Indica conclusão bem-sucedida.

#### **MQCC\_FAILED**

A chamada falhou.

### **Razão**

Tipo: MQLONG - saída

Código de razão qualificando *CompCode*.

Se *CompCode* for MQCC\_OK:

#### **MQRC\_NONE**

(0, X'000') Nenhuma razão a relatar.

Se *CompCode* for MQCC\_FAILED:

#### **MQRC\_NOT\_PRIVILEGED**

(2584, X'A18') Este usuário não é um ID do usuário privilegiado

#### **MQRC\_UNKNOWN\_ENTITY**

(2292, X'8F4') Entidade desconhecida para serviço.

#### **MQRC\_SERVICE\_ERROR**

(2289, X'8F1') Ocorreu um erro inesperado ao acessar o serviço.

#### **MQRC\_SERVICE\_NOT\_AVAILABLE**

(2285, X'8ED') Serviço subjacente não disponível.

Para obter mais informações sobre esses códigos de razão, consulte [conclusão da API e códigos de razão](#).

## Chamada C

```
MQZ_CHECK_PRIVILEGED (QMgrName, &EntityData, EntityType,  
ComponentData, &Continuation,  
&CompCode, &Reason);
```

Os parâmetros transmitidos para o serviço são declarados da seguinte forma:

```
MQCHAR48  QMgrName;           /* Queue manager name */  
MQZED     EntityData;        /* Entity name */  
MQLONG    EntityType;        /* Entity type */  
MQBYTE    ComponentData[n];  /* Component data */  
MQLONG    Continuation;      /* Continuation indicator set by  
                             component */  
MQLONG    CompCode;          /* Completion code */  
MQLONG    Reason;            /* Reason code qualifying CompCode */
```

## MQZ\_COPY\_ALL\_AUTHORITY-Copie todas as autoridades

Essa função é fornecida por um componente de serviço de autorização. É iniciado pelo gerenciador de filas para copiar todas as autorizações que estão atualmente em vigor para um objeto de referência para outro objeto.

O identificador de função para esta função (para MQZEP) é MQZID\_COPY\_ALL\_AUTHORITY.

### Sintaxe

```
MQZ_COPY_ALL_AUTHORITY( QMgrName , RefObjectName , ObjectName , ObjectType ,  
ComponentData , Continuation , CompCode , Reason )
```

### Parâmetros

#### QMgrName

Tipo: MQCHAR48 - entrada

Nome do gerenciador de filas. O nome do gerenciador de fila chamando o componente. Esse nome é preenchido com espaços em branco até o comprimento total do parâmetro; o nome não é terminado por um caractere nulo.

O nome do gerenciador de filas é transmitido para o componente para obter informações; a interface de serviço de autorização não requer que o componente o use de nenhuma maneira definida.

#### Nome do RefObject

Tipo: MQCHAR48 - entrada

Nome do objeto de referência. O nome do objeto de referência, cujas autorizações devem ser copiadas. O comprimento máximo da sequência é de 48 caracteres; se for mais curto do que isso, será preenchido à direita com espaços em branco. O nome não é terminado por um caractere nulo.

#### ObjectName

Tipo: MQCHAR48 - entrada

Nome do objeto. O nome do objeto para o qual os acessos devem ser configurados. O comprimento máximo da sequência é de 48 caracteres; se for mais curto do que isso, será preenchido à direita com espaços em branco. O nome não é terminado por um caractere nulo.

#### ObjectType

Tipo: MQLONG - entrada

Tipo de objeto. O tipo de entidade especificado por *RefObjectName* e *ObjectName*.. Deve ser um dos valores a seguir:

#### MQOT\_AUTH\_INFO

Informações de autenticação..

**MQOT\_CHANNEL**

Canal.

**MQOT\_CLNTCONN\_CHANEXOL**

Canal de conexão do cliente.

**MQOT\_LISTENER**

Ouvinte.

**MQOT\_NAMELIST**

Lista de nomes

**MQOT\_PROCESS**

process definition.

**MQOT\_Q**

Fila.

**MQOT\_Q\_MGR**

Gerenciador de Filas

**MQOT\_SERVICE**

Serviço.

**MQOT\_TOPIC**

:NONE.

**ComponentData**

Tipo: MQBYTEComponentDataLength -entrada/saída.

Dados do componente.. Esses dados são mantidos pelo gerenciador de filas em nome desse componente específico; quaisquer mudanças feitas nele por qualquer uma das funções fornecidas por esse componente são preservados e apresentados na próxima vez em que uma dessas funções do componente for chamada

O comprimento desta área de dados é passado pelo gerenciador de fila no parâmetro de Comprimento ComponentDatada chamada MQZ\_INIT\_AUTHORITY

**Continuação**

Tipo: MQLONG - saída

Indicador de continuação configurado por componente.. Os valores a seguir podem ser especificados:

**MQZCI\_DEFAULT**

Continuação dependente do gerenciador de filas..

Para MQZ\_CHECK\_AUTHORITY, isso tem o mesmo efeito que MQZCI\_STOP

**MQZCI\_CONTINUE**

Continue com o próximo componente

**MQZCI\_STOP**

Não continue com o próximo componente

**CompCode**

Tipo: MQLONG - saída

Código de conclusão. Deve ser um dos valores a seguir:

**MQCC\_OK**

Indica conclusão bem-sucedida.

**MQCC\_FAILED**

A chamada falhou.

**Razão**

Tipo: MQLONG - saída

Código de razão qualificando *CompCode*.

Se *CompCode* for MQCC\_OK:



### **MQRC\_NONE**

(0, X'000') Nenhuma razão a relatar.

Se *CompCode* for MQCC\_FAILED:

### **MQRC\_SERVICE\_ERROR**

(2289, X'8F1') Ocorreu um erro inesperado ao acessar o serviço.

### **MQRC\_SERVICE\_NOT\_AVAILABLE**

(2285, X'8ED') Serviço subjacente não disponível.

### **MQRC\_UNKNOWN\_REF\_OBJECT**

(2294, X'8F6') Objeto de referência desconhecido.

Para obter mais informações sobre esses códigos de razão, consulte [conclusão da API e códigos de razão](#).

## **Chamada C**

```
MQZ_COPY_ALL_AUTHORITY (QMgrName, RefObjectName, ObjectName, ObjectType,  
                        ComponentData, &Continuation, &CompCode,  
                        &Reason);
```

Os parâmetros transmitidos para o serviço são declarados da seguinte forma:

```
MQCHAR48  QMgrName;           /* Queue manager name */  
MQCHAR48  RefObjectName;     /* Reference object name */  
MQCHAR48  ObjectName;       /* Object name */  
MQLONG    ObjectType;        /* Object type */  
MQBYTE    ComponentData[n]; /* Component data */  
MQLONG    Continuation;     /* Continuation indicator set by  
                             component */  
MQLONG    CompCode;         /* Completion code */  
MQLONG    Reason;           /* Reason code qualifying CompCode */
```

## **MQZ\_DELETE\_AUTHORITY-Excluir autoridade**

Essa função é fornecida por um componente de serviço de autorização e é iniciada pelo gerenciador de filas para excluir todas as autorizações associadas ao objeto especificado.

O identificador de função para esta função (para MQZEP) é MQZID\_DELETE\_AUTHORITY.

### **Sintaxe**

```
MQZ_DELETE_AUTHORITY( QMgrName , ObjectName , ObjectType , ComponentData ,  
Continuation , CompCode , Reason )
```

### **Parâmetros**

#### **QMgrName**

Tipo: MQCHAR48 - entrada

Nome do gerenciador de filas. O nome do gerenciador de fila chamando o componente. Esse nome é preenchido com espaços em branco até o comprimento total do parâmetro; o nome não é terminado por um caractere nulo.

O nome do gerenciador de filas é transmitido para o componente para obter informações; a interface de serviço de autorização não requer que o componente o use de nenhuma maneira definida

#### **ObjectName**

Tipo: MQCHAR48 - entrada

Nome do objeto. O nome do objeto para o qual os acessos devem ser excluídos. O comprimento máximo da sequência é de 48 caracteres; se for mais curto do que isso, será preenchido à direita com espaços em branco. O nome não é terminado por um caractere nulo.

Se *ObjectType* for MQOT\_Q\_MGR, esse nome será o mesmo que *QMgrName*

### **ObjectType**

Tipo: MQLONG - entrada

Tipo de objeto. O tipo de entidade especificado por *ObjectName* Deve ser um dos valores a seguir:

#### **MQOT\_AUTH\_INFO**

Informações de autenticação..

#### **MQOT\_CHANNEL**

Canal.

#### **MQOT\_CLNTCONN\_CHANEXOL**

Canal de conexão do cliente.

#### **MQOT\_LISTENER**

Ouvinte.

#### **MQOT\_NAMELIST**

Lista de nomes

#### **MQOT\_PROCESS**

process definition.

#### **MQOT\_Q**

Fila.

#### **MQOT\_Q\_MGR**

Gerenciador de Filas

#### **MQOT\_SERVICE**

Serviço.

#### **MQOT\_TOPIC**

:NONE.

### **ComponentData**

Tipo: MQBYTE x ComponentDataComprimento-entrada/saída

Dados do componente.. Esses dados são mantidos pelo gerenciador de filas em nome desse componente específico; quaisquer mudanças feitas nele por qualquer uma das funções fornecidas por esse componente são preservados e apresentados na próxima vez em que uma dessas funções do componente for chamada

O comprimento desta área de dados é passado pelo gerenciador de fila no parâmetro de Comprimento ComponentData chamada MQZ\_INIT\_AUTHORITY

### **Continuação**

Tipo: MQLONG - saída

Indicador de continuação configurado por componente.. Os valores a seguir podem ser especificados:

#### **MQZCI\_DEFAULT**

Continuação dependente do gerenciador de filas..

Para MQZ\_CHECK\_AUTHORITY, isso tem o mesmo efeito que MQZCI\_STOP

#### **MQZCI\_CONTINUE**

Continue com o próximo componente

#### **MQZCI\_STOP**

Não continue com o próximo componente

### **CompCode**

Tipo: MQLONG - saída

Código de conclusão. Deve ser um dos valores a seguir:

#### **MQCC\_OK**

Indica conclusão bem-sucedida.

## **MQCC\_FAILED**

A chamada falhou.

### **Razão**

Tipo: MQLONG - saída

Código de razão qualificando *CompCode*.

Se *CompCode* for MQCC\_OK:

### **MQRC\_NONE**

(0, X'000') Nenhuma razão a relatar.

Se *CompCode* for MQCC\_FAILED:

### **MQRC\_SERVICE\_ERROR**

(2289, X'8F1') Ocorreu um erro inesperado ao acessar o serviço.

### **MQRC\_SERVICE\_NOT\_AVAILABLE**

(2285, X'8ED') Serviço subjacente não disponível.

Para obter mais informações sobre esses códigos de razão, consulte [conclusão da API e códigos de razão](#).

## **Chamada C**

```
MQZ_DELETE_AUTHORITY (QMgrName, ObjectName, ObjectType, ComponentData,  
&Continuation, &CompCode, &Reason);
```

Os parâmetros transmitidos para o serviço são declarados da seguinte forma:

```
MQCHAR48 QMgrName;          /* Queue manager name */  
MQCHAR48 ObjectName;       /* Object name */  
MQLONG   ObjectType;       /* Object type */  
MQBYTE   ComponentData[n]; /* Component data */  
MQLONG   Continuation;     /* Continuation indicator set by  
                           component */  
MQLONG   CompCode;         /* Completion code */  
MQLONG   Reason;           /* Reason code qualifying CompCode */
```

## **MQZ\_ENUMERATE\_AUTHORITY\_DATA-Enumerar dados de autoridade**

Essa função é fornecida por um componente de serviço de autorização MQZAS\_VERSION\_4 e é iniciada repetidamente pelo gerenciador de filas para recuperar todos os dados de autoridade que correspondem aos critérios de seleção especificados na primeira chamada.

O identificador de função para essa função (para MQZEP) é MQZID\_ENUMERATE\_AUTHORITY\_DATA.

### **Sintaxe**

```
MQZ_ENUMERATE_AUTHORITY_DATA( QMgrName , StartEnumeration , Filter ,  
AuthorityBufferLength , AuthorityBuffer , AuthorityDataLength , ComponentData ,  
Continuation , CompCode , Reason )
```

### **Parâmetros**

#### **QMgrName**

Tipo: MQCHAR48 - entrada

Nome do gerenciador de filas. O nome do gerenciador de fila chamando o componente. Esse nome é preenchido com espaços em branco até o comprimento total do parâmetro; o nome não é terminado por um caractere nulo.

O nome do gerenciador de filas é transmitido para o componente para obter informações; a interface de serviço de autorização não requer que o componente o use de nenhuma maneira definida

## StartEnumeration

Tipo: MQLONG - entrada

Sinalizador que indica se a chamada pode iniciar a enumeração Indica se a chamada pode iniciar a enumeração de dados de autoridade ou continuar a enumeração de dados de autoridade iniciados por uma chamada anterior para MQZ\_ENUMERATE\_AUTHORITY\_DATA. O valor é um dos seguintes valores:

### MQZSE\_START

Inicie a enumeração A chamada é iniciada com este valor para iniciar a enumeradas de dados de autoridade. O parâmetro **Filter** especifica os critérios de seleção a serem usados para selecionar dados de autoridade retornados por esta e chamadas sucessivas.

### MQZSE\_CONTINUE

Continuar a enumeração A chamada é iniciada com este valor para continuar a contagem de dados de autoridade. O parâmetro **Filter** é ignorado neste caso e pode ser especificado como ponteiro nulo (os critérios de seleção são determinados pelo parâmetro **Filter** especificado pela chamada que tinha *StartEnumeration* configurado como MQZSE\_START).

## Filtrar

Tipo: MQZAD-entrada

Filtro. Se *StartEnumeration* for MQZSE\_START, *Filter* especifica os critérios de seleção a serem usados para selecionar os dados de autoridade a serem retornados. Se *Filter* for o ponteiro nulo, nenhum critério de seleção será usado, ou seja, todos os dados de autoridade serão retornados.. Consulte [“MQZAD-Dados de autoridade.” na página 1734](#) para obter detalhes sobre os critérios de seleção que podem ser usados

Se *StartEnumeration* for MQZSE\_CONTINUE, *Filter* será ignorado e poderá ser especificado como o ponteiro nulo.

## Comprimento de AuthorityBuffer

Tipo: MQLONG - entrada

O comprimento de *AuthorityBuffer* Esse é o comprimento em bytes do parâmetro **AuthorityBuffer** . O buffer de autoridade deve ser grande o suficiente para acomodar os dados a serem retornadas

## AuthorityBuffer

Tipo: MQZAD-saída

Dados de autoridade Este é o buffer no qual os dados de autoridade são retornadas O buffer deve ser grande o suficiente para acomodar uma estrutura MQZAD, uma estrutura MQZED, mais o nome da entidade mais longa e o nome de domínio mais longo definido

**Nota:** Nota: Esse parâmetro é definido como um MQZAD, pois o MQZAD sempre ocorre no início do buffer. No entanto, se o buffer for declarado como um MQZAD, o buffer será muito pequeno-ele deve ser maior que um MQZAD para que ele possa acomodar o MQZAD, MQZED, além de nomes de entidade e domínio.

## Comprimento de AuthorityData

Tipo: MQLONG - saída

O comprimento dos dados retornados em *AuthorityBuffer* Se o buffer de autoridade for muito pequeno, o *AuthorityDataLength* será configurado para o comprimento do buffer necessário e a chamada retornará o código de conclusão MQCC\_FAILED e o código de razão MQRC\_BUFFER\_LENGTH\_ERROR.

## ComponentData

Tipo: MQBYTE x ComponentDataComprimento-entrada/saída

Dados do componente.. Esses dados são mantidos pelo gerenciador de filas em nome desse componente específico; quaisquer mudanças feitas nele por qualquer uma das funções fornecidas por esse componente são preservados e apresentados na próxima vez em que uma dessas funções do componente for chamada

O comprimento desta área de dados é passado pelo gerenciador de fila no parâmetro de Comprimento ComponentData chamada MQZ\_INIT\_AUTHORITY

### Continuação

Tipo: MQLONG - saída

Indicador de continuação configurado por componente.. Os valores a seguir podem ser especificados:

#### **MQZCI\_DEFAULT**

Continuação dependente do gerenciador de filas..

Para MQZ\_ENUMERATE\_AUTHORITY\_DATA, isso tem o mesmo efeito que MQZCI\_CONTINUE

#### **MQZCI\_CONTINUE**

Continue com o próximo componente

#### **MQZCI\_STOP**

Não continue com o próximo componente

### CompCode

Tipo: MQLONG - saída

Código de conclusão. Deve ser um dos valores a seguir:

#### **MQCC\_OK**

Indica conclusão bem-sucedida.

#### **MQCC\_FAILED**

A chamada falhou.

### Razão

Tipo: MQLONG - saída

Código de razão qualificando *CompCode*.

Se *CompCode* for MQCC\_OK:

#### **MQRC\_NONE**

(0, X'000') Nenhuma razão a relatar.

Se *CompCode* for MQCC\_FAILED:

#### **MQRC\_BUFFER\_LENGTH\_ERROR**

(2005, X'7D5') Parâmetro de comprimento de buffer inválido.

#### **MQRC\_NO\_DATA\_AVAILABLE**

(2379, X'94B') Não há dados disponíveis.

#### **MQRC\_SERVICE\_ERROR**

(2289, X'8F1') Ocorreu um erro inesperado ao acessar o serviço.

Para obter mais informações sobre esses códigos de razão, consulte [conclusão da API e códigos de razão](#).

## Chamada C

```
MQZ_ENUMERATE_AUTHORITY_DATA (QMgrName, StartEnumeration, &Filter,  
                               AuthorityBufferLength,  
                               &AuthorityBuffer,  
                               &AuthorityDataLength, ComponentData,  
                               &Continuation, &CompCode,  
                               &Reason);
```

Os parâmetros transmitidos para o serviço são declarados da seguinte forma:

```
MQCHAR48  QMgrName;           /* Queue manager name */  
MQLONG    StartEnumeration;   /* Flag indicating whether call should  
                               start enumeration */  
MQZAD     Filter;             /* Filter */  
MQLONG    AuthorityBufferLength; /* Length of AuthorityBuffer */
```

MQZAD	AuthorityBuffer;	/* Authority data */
MQLONG	AuthorityDataLength;	/* Length of data returned in AuthorityBuffer */
MQBYTE	ComponentData[n];	/* Component data */
MQLONG	Continuation;	/* Continuation indicator set by component */
MQLONG	CompCode;	/* Completion code */
MQLONG	Reason;	/* Reason code qualifying CompCode */

## MQZ\_FREE\_USER-Usuário Livre

Essa função é fornecida por um componente de serviço de autorização MQZAS\_VERSION\_5 e é iniciada pelo gerenciador de filas para liberar recurso alocado associado.

É iniciado quando um aplicativo conclui a execução em todos os contextos do usuário, por exemplo, durante uma chamada MQI MQDISC.

O identificador de função para essa função (para MQZEP) é MQZID\_FREE\_USER

### Sintaxe

MQZ\_FREE\_USER( QMgrName , FreeParms , ComponentData , Continuation , CompCode , Reason )

### Parâmetros

#### QMgrName

Tipo: MQCHAR48 - entrada

Nome do gerenciador de filas. O nome do gerenciador de fila chamando o componente. Esse nome é preenchido com espaços em branco até o comprimento total do parâmetro; o nome não é terminado por um caractere nulo.

O nome do gerenciador de filas é transmitido para o componente para obter informações; a interface de serviço de autorização não requer que o componente o use de nenhuma maneira definida

#### FreeParms

Tipo: MQZFP-entrada

Parâmetros gratuitos. Uma estrutura contendo dados relacionados ao recurso a ser liberado. Consulte [“MQZFP-Parâmetros livres”](#) na página 1740 para obter detalhes.

#### ComponentData

Tipo: MQBYTE x ComponentDataComprimento-entrada/saída

Dados do componente.. Esses dados são mantidos pelo gerenciador de filas em nome desse componente específico; quaisquer mudanças feitas nele por qualquer uma das funções fornecidas por esse componente são preservados e apresentados na próxima vez em que uma dessas funções do componente for chamada

O comprimento desta área de dados é passado pelo gerenciador de fila no parâmetro de Comprimento ComponentData chamada MQZ\_INIT\_AUTHORITY

#### Continuação

Tipo: MQLONG - saída

Sinalizador de continuação Os valores a seguir podem ser especificados:

#### MQZCI\_DEFAULT

Continuação dependente de outros componentes..

#### MQZCI\_STOP

Não continue com o próximo componente

#### CompCode

Tipo: MQLONG - saída

Código de conclusão. Deve ser um dos valores a seguir:

## **MQCC\_OK**

Indica conclusão bem-sucedida.

## **MQCC\_FAILED**

A chamada falhou.

## **Razão**

Tipo: MQLONG - saída

Código de razão qualificando *CompCode*.

Se *CompCode* for MQCC\_OK:

## **MQRC\_NONE**

(0, X'000') Nenhuma razão a relatar.

Se *CompCode* for MQCC\_FAILED:

## **MQRC\_SERVICE\_ERROR**

(2289, X'8F1') Ocorreu um erro inesperado ao acessar o serviço.

Para obter mais informações sobre esses códigos de razão, consulte [conclusão da API e códigos de razão](#).

## **Chamada C**

```
MQZ_AUTHENTICATE_USER (QMgrName, SecurityParms, ApplicationContext,  
IdentityContext, CorrelationPtr, ComponentData,  
&Continuation, &CompCode, &Reason);
```

Os parâmetros transmitidos para o serviço são declarados da seguinte forma:

```
MQCHAR48  QMgrName;           /* Queue manager name */  
MQZFP     FreeParms;         /* Resource to be freed */  
MQBYTE    ComponentData[n]; /* Component data */  
MQLONG    Continuation;     /* Continuation indicator set by  
                           component */  
MQLONG    CompCode;         /* Completion code */  
MQLONG    Reason;          /* Reason code qualifying CompCode */
```

## **MQZ\_GET\_AUTHORITY-Obter autoridade**

Essa função é fornecida por um componente de serviço de autorização MQZAS\_VERSION\_1 e é iniciada pelo gerenciador de filas para recuperar a autoridade que uma entidade tem para acessar o objeto especificado, incluindo (se a entidade for um principal) autoridades possuídas pelos grupos nos quais o principal é um membro.. As autoridades de perfis genéricos são incluídas no conjunto de autoridades retornado

O identificador de função para essa função (para MQZEP) é MQZID\_GET\_AUTHORITY

## **Sintaxe**

```
MQZ_GET_AUTHORITY( QMgrName , EntityName , EntityType , ObjectName ,  
ObjectType , Authority , ComponentData , Continuation , CompCode , Reason )
```

## **Parâmetros**

### **QMgrName**

Tipo: MQCHAR48 - entrada

Nome do gerenciador de filas. O nome do gerenciador de fila chamando o componente. Esse nome é preenchido com espaços em branco até o comprimento total do parâmetro; o nome não é terminado por um caractere nulo.

O nome do gerenciador de filas é transmitido para o componente para obter informações; a interface de serviço de autorização não requer que o componente o use de nenhuma maneira definida

### **EntityName**

Tipo: MQCHAR12 - entrada

Nome da entidade.. O nome da entidade cujo acesso ao objeto deve ser recuperado. O comprimento máximo da sequência é de 12 caracteres; se ela for mais curta do que ela for preenchida à direita com espaços em branco. O nome não é terminado por um caractere nulo.

### **EntityType**

Tipo: MQLONG - entrada

Tipo de entidade.. O tipo de entidade especificado por *EntityName* Deve ser um dos valores a seguir:

#### **MQZAET\_PRINCIPAL**

Principal.

#### **MQZAET\_GROUP**

.

### **ObjectName**

Tipo: MQCHAR48 - entrada

Nome do objeto. O nome do objeto para o qual o acesso deve ser recuperado O comprimento máximo da sequência é de 48 caracteres; se for mais curto do que isso, será preenchido à direita com espaços em branco. O nome não é terminado por um caractere nulo.

Se *ObjectType* for MQOT\_Q\_MGR, esse nome será o mesmo que *QMgrName*

### **ObjectType**

Tipo: MQLONG - entrada

Tipo de objeto. O tipo de entidade especificado por *ObjectName* Deve ser um dos valores a seguir:

#### **MQOT\_AUTH\_INFO**

Informações de autenticação..

#### **MQOT\_CHANNEL**

Canal.

#### **MQOT\_CLNTCONN\_CHANEXOL**

Canal de conexão do cliente.

#### **MQOT\_LISTENER**

Ouvinte.

#### **MQOT\_NAMELIST**

Lista de nomes

#### **MQOT\_PROCESS**

process definition.

#### **MQOT\_Q**

Fila.

#### **MQOT\_Q\_MGR**

Gerenciador de Filas

#### **MQOT\_SERVICE**

Serviço.

#### **MQOT\_TOPIC**

:NONE.

### **Autoridade**

Tipo: MQLONG - entrada

Autoridade da entidade.. Se a entidade tiver uma autoridade, esse campo será igual à operação de autorização apropriada (constante MQZAO\_\*).. Se tiver mais de uma autoridade, esse campo será o OR bit a bit das constantes MQZAO\_\* correspondentes.



## ComponentData

Tipo: MQBYTE xComponentDataComprimento-entrada/saída

Dados do componente.. Esses dados são mantidos pelo gerenciador de filas em nome desse componente específico; quaisquer mudanças feitas nele por qualquer uma das funções fornecidas por esse componente são preservados e apresentados na próxima vez em que uma dessas funções do componente for chamada

O comprimento dessa área de dados é passado pelo gerenciador de filas no parâmetro **ComponentDataLength** da chamada MQZ\_INIT\_AUTHORITY

## Continuação

Tipo: MQLONG - saída

Indicador de continuação configurado por componente.. Os valores a seguir podem ser especificados:

### **MQZCI\_DEFAULT**

Continuação dependente do gerenciador de filas..

Para MQZ\_GET\_AUTHORITY, isso tem o mesmo efeito que MQZCI\_CONTINUE

### **MQZCI\_CONTINUE**

Continue com o próximo componente

### **MQZCI\_STOP**

Não continue com o próximo componente

## CompCode

Tipo: MQLONG - saída

Código de conclusão. Deve ser um dos valores a seguir:

### **MQCC\_OK**

Indica conclusão bem-sucedida.

### **MQCC\_FAILED**

A chamada falhou.

## Razão

Tipo: MQLONG - saída

Código de razão qualificando *CompCode*.

Se *CompCode* for MQCC\_OK:

### **MQRC\_NONE**

(0, X'000') Nenhuma razão a relatar.

Se *CompCode* for MQCC\_FAILED:

### **MQRC\_NOT\_AUTHORIZED**

(2035, X'7F3') Não autorizado para acesso.

### **MQRC\_SERVICE\_ERROR**

(2289, X'8F1') Ocorreu um erro inesperado ao acessar o serviço.

### **MQRC\_SERVICE\_NOT\_AVAILABLE**

(2285, X'8ED') Serviço subjacente não disponível.

### **MQRC\_UNKNOWN\_ENTITY**

(2292, X'8F4') Entidade desconhecida para serviço.

Para obter mais informações sobre esses códigos de razão, consulte [conclusão da API e códigos de razão](#).

## Chamada C

```
MQZ_GET_AUTHORITY (QMgzName, EntityName, EntityType, ObjectName,
```

```
ObjectType, &Authority, ComponentData,  
&Continuation, &CompCode, &Reason);
```

Os parâmetros transmitidos para o serviço são declarados da seguinte forma:

```
MQCHAR48  QMgrName;           /* Queue manager name */  
MQCHAR12  EntityName;        /* Entity name */  
MQLONG    EntityType;        /* Entity type */  
MQCHAR48  ObjectName;        /* Object name */  
MQLONG    ObjectType;        /* Object type */  
MQLONG    Authority;         /* Authority of entity */  
MQBYTE    ComponentData[n]; /* Component data */  
MQLONG    Continuation;      /* Continuation indicator set by  
                             component */  
MQLONG    CompCode;          /* Completion code */  
MQLONG    Reason;            /* Reason code qualifying CompCode */
```

## MQZ\_GET\_AUTHORITY\_2 -Obter autoridade (estendido)

Essa função é fornecida por um componente de serviço de autorização MQZAS\_VERSION\_2 e é iniciada pelo gerenciador de filas para recuperar a autoridade que uma entidade tem para acessar o objeto especificado.

O identificador de função para essa função (para MQZEP) é MQZID\_GET\_AUTHORITY

MQZ\_GET\_AUTHORITY\_2 é como MQZ\_GET\_AUTHORITY, mas com o parâmetro **EntityName** substituído pelo parâmetro **EntityData**.

### Sintaxe

```
MQZ_GET_AUTHORITY_2( QMgrName , EntityData , EntityType , ObjectName ,  
ObjectType , Authority , ComponentData , Continuation , CompCode , Reason )
```

### Parâmetros

#### QMgrName

Tipo: MQCHAR48 - entrada

Nome do gerenciador de filas. O nome do gerenciador de fila chamando o componente. Esse nome é preenchido com espaços em branco até o comprimento total do parâmetro; o nome não é terminado por um caractere nulo.

O nome do gerenciador de filas é transmitido para o componente para obter informações; a interface de serviço de autorização não requer que o componente o use de nenhuma maneira definida

#### EntityData

Tipo: MQZED-entrada

Dados da entidade.. Dados relacionados à entidade para a qual a autorização para o objeto será recuperada. Consulte [“MQZED-Descritor de entidade”](#) na página 1737 para obter detalhes.

#### EntityType

Tipo: MQLONG - entrada

Tipo de entidade.. O tipo de entidade especificado por *EntityData* Deve ser um dos valores a seguir:

#### MQZAET\_PRINCIPAL

Principal.

#### MQZAET\_GROUP

.

#### ObjectName

Tipo: MQCHAR48 - entrada

Nome do objeto. O nome do objeto para o qual a autoridade da entidade deve ser recuperada O comprimento máximo da sequência é de 48 caracteres; se for mais curto do que isso, será preenchido à direita com espaços em branco. O nome não é terminado por um caractere nulo.

Se *ObjectType* for MQOT\_Q\_MGR, esse nome será o mesmo que *QMgrName*

### **ObjectType**

Tipo: MQLONG - entrada

Tipo de objeto. O tipo de entidade especificado por *ObjectName* Deve ser um dos valores a seguir:

#### **MQOT\_AUTH\_INFO**

Informações de autenticação..

#### **MQOT\_CHANNEL**

Canal.

#### **MQOT\_CLNTCONN\_CHANEXOL**

Canal de conexão do cliente.

#### **MQOT\_LISTENER**

Ouvinte.

#### **MQOT\_NAMELIST**

Lista de nomes

#### **MQOT\_PROCESS**

process definition.

#### **MQOT\_Q**

Fila.

#### **MQOT\_Q\_MGR**

Gerenciador de Filas

#### **MQOT\_SERVICE**

Serviço.

#### **MQOT\_TOPIC**

:NONE.

### **Autoridade**

Tipo: MQLONG - entrada

Autoridade da entidade.. Se a entidade tiver uma autoridade, esse campo será igual à operação de autorização apropriada (constante MQZAO\_\*).. Se tiver mais de uma autoridade, esse campo será o OR bit a bit das constantes MQZAO\_\* correspondentes.

### **ComponentData**

Tipo: MQBYTE xComponentDataComprimento-entrada/saída

Dados do componente.. Esses dados são mantidos pelo gerenciador de filas em nome desse componente específico; quaisquer mudanças feitas nele por qualquer uma das funções fornecidas por esse componente são preservados e apresentados na próxima vez em que uma dessas funções do componente for chamada

O comprimento dessa área de dados é passado pelo gerenciador de filas no parâmetro

**ComponentDataLength** da chamada MQZ\_INIT\_AUTHORITY

### **Continuação**

Tipo: MQLONG - saída

Indicador de continuação configurado por componente.. Os valores a seguir podem ser especificados:

#### **MQZCI\_DEFAULT**

Continuação dependente do gerenciador de filas..

Para MQZ\_CHECK\_AUTHORITY, isso tem o mesmo efeito que MQZCI\_STOP

#### **MQZCI\_CONTINUE**

Continue com o próximo componente

## **MQZCI\_STOP**

Não continue com o próximo componente

### **CompCode**

Tipo: MQLONG - saída

Código de conclusão. Deve ser um dos valores a seguir:

### **MQCC\_OK**

Indica conclusão bem-sucedida.

### **MQCC\_FAILED**

A chamada falhou.

### **Razão**

Tipo: MQLONG - saída

Código de razão qualificando *CompCode*.

Se *CompCode* for MQCC\_OK:

### **MQRC\_NONE**

(0, X'000') Nenhuma razão a relatar.

Se *CompCode* for MQCC\_FAILED:

### **MQRC\_NOT\_AUTHORIZED**

(2035, X'7F3') Não autorizado para acesso.

### **MQRC\_SERVICE\_ERROR**

(2289, X'8F1') Ocorreu um erro inesperado ao acessar o serviço.

### **MQRC\_SERVICE\_NOT\_AVAILABLE**

(2285, X'8ED') Serviço subjacente não disponível.

### **MQRC\_UNKNOWN\_ENTITY**

(2292, X'8F4') Entidade desconhecida para serviço.

Para obter mais informações sobre esses códigos de razão, consulte [conclusão da API e códigos de razão](#).

## **Chamada C**

```
MQZ_GET_AUTHORITY_2 (QMgrName, &EntityData, EntityType, ObjectName,  
                    ObjectType, &Authority, ComponentData,  
                    &Continuation, &CompCode, &Reason);
```

Os parâmetros transmitidos para o serviço são declarados da seguinte forma:

```
MQCHAR48  QMgrName;           /* Queue manager name */  
MQZED     EntityData;        /* Entity data */  
MQLONG    EntityType;        /* Entity type */  
MQCHAR48  ObjectName;        /* Object name */  
MQLONG    ObjectType;        /* Object type */  
MQLONG    Authority;         /* Authority of entity */  
MQBYTE    ComponentData[n]; /* Component data */  
MQLONG    Continuation;      /* Continuation indicator set by  
                             component */  
MQLONG    CompCode;          /* Completion code */  
MQLONG    Reason;            /* Reason code qualifying CompCode */
```

## **MQZ\_GET\_EXPLICIT\_AUTHORITY-Obter autoridade explícita**

Essa função é fornecida por um componente de serviço de autorização MQZAS\_VERSION\_1 e é iniciada pelo gerenciador de filas para recuperar a autoridade que uma entidade tem para acessar o objeto especificado, incluindo (se a entidade for um principal) autoridades possuídas pelos grupos nos quais o principal é um membro.. As autoridades de perfis genéricos são incluídas no conjunto de autoridades retornado

No AIX and Linux, para o IBM MQ object authority manager (OAM) integrado, a autoridade retornada é aquela possuída somente pelo grupo principal do proprietário.

O identificador de função para essa função (para MQZEP) é MQZID\_GET\_EXPLICIT\_AUTHORITY

## Sintaxe

MQZ\_GET\_EXPLICIT\_AUTHORITY( QMgrName , EntityName , EntityType , ObjectName , ObjectType , Authority , ComponentData , Continuation , CompCode , Reason )

## Parâmetros

### QMgrName

Tipo: MQCHAR48 - entrada

Nome do gerenciador de filas. O nome do gerenciador de fila chamando o componente. Esse nome é preenchido com espaços em branco até o comprimento total do parâmetro; o nome não é terminado por um caractere nulo.

O nome do gerenciador de filas é transmitido para o componente para obter informações; a interface de serviço de autorização não requer que o componente o use de nenhuma maneira definida

### EntityName

Tipo: MQCHAR12 - entrada

Nome da entidade.. O nome da entidade para a qual o acesso ao objeto será recuperado. O comprimento máximo da sequência é de 12 caracteres; se ela for mais curta do que ela for preenchida à direita com espaços em branco. O nome não é terminado por um caractere nulo.

### EntityType

Tipo: MQLONG - entrada

Tipo de entidade.. O tipo de entidade especificado por *EntityName* Deve ser um dos valores a seguir:

#### **MQZAET\_PRINCIPAL**

Principal.

#### **MQZAET\_GROUP**

.

### ObjectName

Tipo: MQCHAR48 - entrada

Nome do objeto. O nome do objeto para o qual a autoridade da entidade deve ser recuperada O comprimento máximo da sequência é de 48 caracteres; se for mais curto do que isso, será preenchido à direita com espaços em branco. O nome não é terminado por um caractere nulo.

Se *ObjectType* for MQOT\_Q\_MGR, esse nome será o mesmo que *QMgrName*

### ObjectType

Tipo: MQLONG - entrada

Tipo de objeto. O tipo de entidade especificado por *ObjectName* Deve ser um dos valores a seguir:

#### **MQOT\_AUTH\_INFO**

Informações de autenticação..

#### **MQOT\_CHANNEL**

Canal.

#### **MQOT\_CLNTCONN\_CHANEXO L**

Canal de conexão do cliente.

#### **MQOT\_LISTENER**

Ouvinte.

#### **MQOT\_NAMELIST**

Lista de nomes

**MQOT\_PROCESS**

process definition.

**MQOT\_Q**

Fila.

**MQOT\_Q\_MGR**

Gerenciador de Filas

**MQOT\_SERVICE**

Serviço.

**MQOT\_TOPIC**

:NONE.

**Autoridade**

Tipo: MQLONG - entrada

Autoridade da entidade.. Se a entidade tiver uma autoridade, esse campo será igual à operação de autorização apropriada (constante MQZAO\_\*).. Se tiver mais de uma autoridade, esse campo será o OR bit a bit das constantes MQZAO\_\* correspondentes.

**ComponentData**

Tipo: MQBYTE x ComponentDataComprimento-entrada/saída

Dados do componente.. Esses dados são mantidos pelo gerenciador de filas em nome desse componente específico; quaisquer mudanças feitas nele por qualquer uma das funções fornecidas por esse componente são preservados e apresentados na próxima vez em que uma dessas funções do componente for chamada

O comprimento dessa área de dados é passado pelo gerenciador de filas no parâmetro **ComponentDataLength** da chamada MQZ\_INIT\_AUTHORITY

**Continuação**

Tipo: MQLONG - saída

Indicador de continuação configurado por componente.. Os valores a seguir podem ser especificados:

**MQZCI\_DEFAULT**

Continuação dependente do gerenciador de filas..

Para MQZ\_GET\_AUTHORITY, isso tem o mesmo efeito que MQZCI\_CONTINUE

**MQZCI\_CONTINUE**

Continue com o próximo componente

**MQZCI\_STOP**

Não continue com o próximo componente

**CompCode**

Tipo: MQLONG - saída

Código de conclusão. Deve ser um dos valores a seguir:

**MQCC\_OK**

Indica conclusão bem-sucedida.

**MQCC\_FAILED**

A chamada falhou.

**Razão**

Tipo: MQLONG - saída

Código de razão qualificando *CompCode*.

Se *CompCode* for MQCC\_OK:

**MQRC\_NONE**

(0, X'000') Nenhuma razão a relatar.

Se *CompCode* for MQCC\_FAILED:

### **MQRC\_NOT\_AUTHORIZED**

(2035, X'7F3') Não autorizado para acesso.

### **MQRC\_SERVICE\_ERROR**

(2289, X'8F1') Ocorreu um erro inesperado ao acessar o serviço.

### **MQRC\_SERVICE\_NOT\_AVAILABLE**

(2285, X'8ED') Serviço subjacente não disponível.

### **MQRC\_UNKNOWN\_ENTITY**

(2292, X'8F4') Entidade desconhecida para serviço.

Para obter mais informações sobre esses códigos de razão, consulte [conclusão da API e códigos de razão](#).

## **Chamada C**

```
MQZ_GET_EXPLICIT_AUTHORITY (QMGrName, EntityName, EntityType,  
                             ObjectName, ObjectType, &Authority,  
                             ComponentData, &Continuation,  
                             &CompCode, &Reason);
```

Os parâmetros transmitidos para o serviço são declarados da seguinte forma:

```
MQCHAR48  QMGrName;           /* Queue manager name */  
MQCHAR12  EntityName;        /* Entity name */  
MQLONG    EntityType;        /* Entity type */  
MQCHAR48  ObjectName;       /* Object name */  
MQLONG    ObjectType;       /* Object type */  
MQLONG    Authority;        /* Authority of entity */  
MQBYTE    ComponentData[n]; /* Component data */  
MQLONG    Continuation;     /* Continuation indicator set by  
                             component */  
MQLONG    CompCode;         /* Completion code */  
MQLONG    Reason;           /* Reason code qualifying CompCode */
```

## **MQZ\_GET\_EXPLICIT\_AUTHORITY\_2 - Obter autoridade explícita (estendida)**

Essa função é fornecida por um componente de serviço de autorização MQZAS\_VERSION\_2 e é iniciada pelo gerenciador de filas para recuperar a autoridade que um grupo nomeado tem para acessar um objeto especificado (mas sem a autoridade adicional do grupo **nobody**) ou a autoridade que o grupo primário do proprietário nomeado tem para acessar um objeto especificado.

O identificador de função para essa função (para MQZEP) é MQZID\_GET\_EXPLICIT\_AUTHORITY

MQZ\_GET\_EXPLICIT\_AUTHORITY\_2 é como MQZ\_GET\_EXPLICIT\_AUTHORITY, mas com o parâmetro **EntityName** substituído pelo parâmetro **EntityData**.

### **Sintaxe**

```
MQZ_GET_EXPLICIT_AUTHORITY_2( QMGrName , EntityData , EntityType , ObjectName ,  
ObjectType , Authority , ComponentData , Continuation , CompCode , Reason )
```

### **Parâmetros**

#### **QMGrName**

Tipo: MQCHAR48 - entrada

Nome do gerenciador de filas. O nome do gerenciador de fila chamando o componente. Esse nome é preenchido com espaços em branco até o comprimento total do parâmetro; o nome não é terminado por um caractere nulo.

O nome do gerenciador de filas é transmitido para o componente para obter informações; a interface de serviço de autorização não requer que o componente o use de nenhuma maneira definida

**EntityData**

Tipo: MQZED-entrada

Dados da entidade.. Dados relacionados à entidade cuja autorização para o objeto deve ser recuperado. Consulte [“MQZED-Descrição de entidade”](#) na página 1737 para obter detalhes.

**EntityType**

Tipo: MQLONG - entrada

Tipo de entidade.. O tipo de entidade especificado por *EntityData* Deve ser um dos valores a seguir:

**MQZAET\_PRINCIPAL**

Principal.

**MQZAET\_GROUP****ObjectName**

Tipo: MQCHAR48 - entrada

Nome do objeto. O nome do objeto para o qual a autoridade da entidade deve ser recuperada O comprimento máximo da sequência é de 48 caracteres; se for mais curto do que isso, será preenchido à direita com espaços em branco. O nome não é terminado por um caractere nulo.

Se *ObjectType* for MQOT\_Q\_MGR, esse nome será o mesmo que *QMgrName*

**ObjectType**

Tipo: MQLONG - entrada

Tipo de objeto. O tipo de entidade especificado por *ObjectName* Deve ser um dos valores a seguir:

**MQOT\_AUTH\_INFO**

Informações de autenticação..

**MQOT\_CHANNEL**

Canal.

**MQOT\_CLNTCONN\_CHANEXO L**

Canal de conexão do cliente.

**MQOT\_LISTENER**

Ouvinte.

**MQOT\_NAMELIST**

Lista de nomes

**MQOT\_PROCESS**

process definition.

**MQOT\_Q**

Fila.

**MQOT\_Q\_MGR**

Gerenciador de Filas

**MQOT\_SERVICE**

Serviço.

**MQOT\_TOPIC**

:NONE.

**Autoridade**

Tipo: MQLONG - entrada

Autoridade da entidade.. Se a entidade tiver uma autoridade, esse campo será igual à operação de autorização apropriada (constante MQZAO\_\*).. Se tiver mais de uma autoridade, esse campo será o OR bit a bit das constantes MQZAO\_\* correspondentes.

**ComponentData**

Tipo: MQBYTE xComponentDataComprimento-entrada/saída



Dados do componente.. Esses dados são mantidos pelo gerenciador de filas em nome desse componente específico; quaisquer mudanças feitas nele por qualquer uma das funções fornecidas por esse componente são preservados e apresentados na próxima vez em que uma dessas funções do componente for chamada

O comprimento dessa área de dados é passado pelo gerenciador de filas no parâmetro **ComponentDataLength** da chamada MQZ\_INIT\_AUTHORITY

### Continuação

Tipo: MQLONG - saída

Indicador de continuação configurado por componente.. Os valores a seguir podem ser especificados:

#### **MQZCI\_DEFAULT**

Continuação dependente do gerenciador de filas..

Para MQZ\_CHECK\_AUTHORITY, isso tem o mesmo efeito que MQZCI\_STOP

#### **MQZCI\_CONTINUE**

Continue com o próximo componente

#### **MQZCI\_STOP**

Não continue com o próximo componente

### CompCode

Tipo: MQLONG - saída

Código de conclusão. Deve ser um dos valores a seguir:

#### **MQCC\_OK**

Indica conclusão bem-sucedida.

#### **MQCC\_FAILED**

A chamada falhou.

### Razão

Tipo: MQLONG - saída

Código de razão qualificando *CompCode*.

Se *CompCode* for MQCC\_OK:

#### **MQRC\_NONE**

(0, X'000') Nenhuma razão a relatar.

Se *CompCode* for MQCC\_FAILED:

#### **MQRC\_NOT\_AUTHORIZED**

(2035, X'7F3') Não autorizado para acesso.

#### **MQRC\_SERVICE\_ERROR**

(2289, X'8F1') Ocorreu um erro inesperado ao acessar o serviço.

#### **MQRC\_SERVICE\_NOT\_AVAILABLE**

(2285, X'8ED') Serviço subjacente não disponível.

#### **MQRC\_UNKNOWN\_ENTITY**

(2292, X'8F4') Entidade desconhecida para serviço.

Para obter mais informações sobre esses códigos de razão, consulte [conclusão da API e códigos de razão](#).

## Chamada C

```
MQZ_GET_EXPLICIT_AUTHORITY_2 (QMgrName, &EntityData, EntityType,  
                               ObjectName, ObjectType, &Authority,  
                               ComponentData, &Continuation,  
                               &CompCode, &Reason);
```

Os parâmetros transmitidos para o serviço são declarados da seguinte forma:

```
MQCHAR48  QMgrName;           /* Queue manager name */
MQZED      EntityData;       /* Entity data */
MQLONG     EntityType;       /* Entity type */
MQCHAR48   ObjectName;       /* Object name */
MQLONG     ObjectType;       /* Object type */
MQLONG     Authority;        /* Authority of entity */
MQBYTE     ComponentData[n]; /* Component data */
MQLONG     Continuation;     /* Continuation indicator set by
                             component */
MQLONG     CompCode;         /* Completion code */
MQLONG     Reason;          /* Reason code qualifying CompCode */
```

## MQZ\_INIT\_AUTHORITY-Inicializar serviço de autorização

Esta função é fornecida por um componente de serviço de autorização e é iniciada pelo gerenciador de filas durante a configuração do componente. É esperado chamar MQZEP para fornecer informações para o gerenciador de filas.

O identificador de função para esta função (para MQZEP) é MQZID\_INIT\_AUTHORITY.

### Sintaxe

`MQZ_INIT_AUTHORITY( Hconfig , Options , QMgrName , ComponentDataLength , ComponentData , Version , CompCode , Reason )`

### Parâmetros

#### Hconfig

Tipo: MQHCONFIG-entrada

Identificador de configuração. Essa manipulação representa o componente específico sendo inicializado. Ele deve ser usado pelo componente ao chamar o gerenciador de filas com a função MQZEP.

#### Opções

Tipo: MQLONG - entrada

Opções de inicialização.. Deve ser um dos valores a seguir:

##### **MQZIO\_PRIMARY**

Inicialização primária..

##### **MQZIO\_SECONDARY**

Inicialização secundária..

#### QMgrName

Tipo: MQCHAR48 - entrada

Nome do gerenciador de filas. O nome do gerenciador de fila chamando o componente. Esse nome é preenchido com espaços em branco até o comprimento total do parâmetro; o nome não é terminado por um caractere nulo.

O nome do gerenciador de filas é transmitido para o componente para obter informações; a interface de serviço de autorização não requer que o componente o use de nenhuma maneira definida.

#### Comprimento de ComponentData

Tipo: MQLONG - entrada

Comprimento dos dados do componente. Comprimento em bytes da área *ComponentData*. Esse comprimento é definido no componente de dados de configuração.

#### ComponentData

Tipo: MQBYTE x ComponentDataComprimento-entrada/saída

Dados do componente.. É inicializado para todos os zeros antes de chamar a função de inicialização primária do componente. Esses dados são mantidos pelo gerenciador de filas em nome desse componente específico; quaisquer mudanças feitas nele por qualquer uma das funções (incluindo a função de inicialização) fornecidas por esse componente são preservadas e apresentadas na próxima vez em que uma dessas funções do componente for chamada.

O comprimento dessa área de dados é passado pelo gerenciador de filas no parâmetro **ComponentDataLength** da chamada MQZ\_INIT\_AUTHORITY

### Versão

Tipo: MQLONG-entrada/saída

Número da versão.. Na entrada para a função de inicialização, isso identifica o número da versão mais alta que o gerenciador de fila suporta.. A função de inicialização deve alterar isso, se necessário, para a versão da interface que ela suporta.. Se no retorno o gerenciador de filas não suportar a versão retornada pelo componente, ele chamará a função MQZ\_TERM\_AUTHORITY do componente e não fará uso adicional desse componente.

Os seguintes valores são suportados:

#### **MQZAS\_VERSION\_1**

Versão 1.

#### **MQZAS\_VERSION\_2**

Versão 2.

#### **MQZAS\_VERSION\_3**

Versão 3.

#### **MQZAS\_VERSION\_4**

Versão 4.

#### **MQZAS\_VERSION\_5**

Versão 5.

#### **MQZAS\_VERSION\_6**

IBM WebSphere MQ 6.

### CompCode

Tipo: MQLONG - saída

Código de conclusão. Deve ser um dos valores a seguir:

#### **MQCC\_OK**

Indica conclusão bem-sucedida.

#### **MQCC\_FAILED**

A chamada falhou.

### Razão

Tipo: MQLONG - saída

Código de razão qualificando *CompCode*.

Se *CompCode* for MQCC\_OK:

#### **MQRC\_NONE**

(0, X'000') Nenhuma razão a relatar.

Se *CompCode* for MQCC\_FAILED:

#### **MQRC\_INITIALIZATION\_FAILED**

(2286, X'8EE') A inicialização falhou por uma razão indefinida..

#### **MQRC\_SERVICE\_NOT\_AVAILABLE**

(2285, X'8ED') Serviço subjacente não disponível.

Para obter mais informações sobre esses códigos de razão, consulte [conclusão da API e códigos de razão](#).

## Chamada C

```
MQZ_INIT_AUTHORITY (Hconfig, Options, QMgrName, ComponentDataLength,  
                    ComponentData, &Version, &CompCode,  
                    &Reason);
```

Os parâmetros transmitidos para o serviço são declarados da seguinte forma:

```
MQHCONFIG  Hconfig;           /* Configuration handle */  
MQLONG     Options;          /* Initialization options */  
MQCHAR48   QMgrName;        /* Queue manager name */  
MQLONG     ComponentDataLength; /* Length of component data */  
MQBYTE     ComponentData[n]; /* Component data */  
MQLONG     Version;         /* Version number */  
MQLONG     CompCode;        /* Completion code */  
MQLONG     Reason;         /* Reason code qualifying CompCode */
```

## MQZ\_INQUIRE-Serviço de autorização de consulta

Essa função é fornecida por um componente de serviço de autorização MQZAS\_VERSION\_5 e é iniciada pelo gerenciador de filas para consultar a funcionalidade suportada..

Quando diversos componentes de serviço são usados, os componentes de serviço são chamados em ordem reversa para a ordem em que eles foram instalados.

O identificador de função para esta função (para MQZEP) é MQZID\_INQUIRE.

### Sintaxe

```
MQZ_INQUIRE( QMgrName , SelectorCount , Selectors , IntAttrCount , IntAttrs ,  
CharAttrLength , CharAttrs , SelectorReturned , ComponentData , Continuation ,  
CompCode , Reason )
```

### Parâmetros

#### QMgrName

Tipo: MQCHAR48 - entrada

Nome do gerenciador de filas. O nome do gerenciador de fila chamando o componente. Esse nome é preenchido com espaços em branco até o comprimento total do parâmetro; o nome não é terminado por um caractere nulo.

O nome do gerenciador de filas é transmitido para o componente para obter informações; a interface de serviço de autorização não requer que o componente o use de nenhuma maneira definida

#### SelectorCount

Tipo: MQLONG - entrada

Número de seletores.. O número de seletores fornecidos no parâmetro **Selectors**

O valor deve estar no intervalo de 0 a 256.

#### Seletores

Tipo: MQLONGxSelectorContagem-entrada

Matriz de seletores.. Cada seletor identifica um atributo necessário e deve ser um dos seguintes:

- MQIACF\_INTERFACE\_VERSION (número inteiro)
- MQIACF\_USER\_ID\_SUPPORT (número inteiro)
- MQCACF\_SERVICE\_COMPONENT (caractere)

Os seletores podem ser especificados em qualquer ordem O número de seletores na matriz é indicado pelo parâmetro **SelectorCount** ..

Os atributos de número inteiro identificados por seletores são retornados no parâmetro **IntAttrs** na mesma ordem em que aparecem em *Selectors*.

Os atributos de caractere identificados por seletores são retornados no parâmetro **CharAttrs** na mesma ordem em que eles aparecem *Selectors*

### **IntAttrCount**

Tipo: MQLONG - entrada

Número de atributos de número inteiro fornecidos no parâmetro IntAttrs .

O valor deve estar no intervalo de 0 a 256.

### **IntAttrs**

Tipo: MQLONG x IntAttrCount-output

Atributos de número inteiro.. Matriz de atributos de número inteiro Os atributos de número inteiro são retornados na mesma ordem que os seletores de número inteiro correspondentes na matriz *Selectors*

### **CharAttrContagem**

Tipo: MQLONG - entrada

Comprimento do buffer de atributo de caractere O comprimento em bytes do parâmetro **CharAttrs** .

O valor deve ser pelo menos a soma dos comprimento dos atributos de caractere solicitados. Se nenhum atributo de caractere for solicitado, zero será um valor válido.

### **CharAttrs**

Tipo: MQLONG x CharAttrContagem-saída

Buffer de atributos de caractere Buffer contendo atributos de caracteres concatenados juntos. Os atributos de caractere são retornados na mesma ordem que os seletores de caracteres correspondentes na matriz *Selectors*

O comprimento do buffer é fornecido pelo parâmetro de contagem CharAttr.

### **SelectorReturned**

Tipo: MQLONG x SelectorCount -entrada

Seletor retornado. Matriz de valores que identificam quais atributos foram retornados do conjunto solicitado pelos seletores no parâmetro *Selectors* O número de valores nessa matriz é indicado pelo parâmetro **SelectorCount** .. Cada valor na matriz está relacionado ao seletor da posição correspondente na matriz *Selectors*. Cada valor é um dos seguintes:

#### **MQZSL\_RETORNADO**

O atributo solicitado pelo seletor correspondente no parâmetro **Selectors** foi retornado..

#### **MQZSL\_NOT\_RETORNADO**

O atributo solicitado pelo seletor correspondente no parâmetro **Selectors** não foi retornado..

A matriz é inicializada com todos os valores como *MQZSL\_NOT\_RETURNED* Quando um componente de serviço de autorização retornar um atributo, ele configurará o valor apropriado na matriz como *MQZSL\_NOT\_RETURNED*. Isso permite que qualquer outro componente de serviço de autorização, para o qual a chamada de consulta é feita, identifique quais atributos já foram retornadas..

### **ComponentData**

Tipo: MQBYTE x ComponentDataComprimento-entrada/saída

Dados do componente.. Esses dados são mantidos pelo gerenciador de filas em nome desse componente específico; quaisquer mudanças feitas nele por qualquer uma das funções fornecidas por esse componente são preservados e apresentados na próxima vez em que uma dessas funções do componente for chamada

O comprimento dessa área de dados é passado pelo gerenciador de filas no parâmetro

**ComponentDataLength** da chamada MQZ\_INIT\_AUTHORITY

### **Continuação**

Tipo: MQLONG - saída

Indicador de continuação configurado por componente.. Os valores a seguir podem ser especificados:

#### **MQZCI\_DEFAULT**

Continuação dependente do gerenciador de filas..

Para MQZ\_CHECK\_AUTHORITY, isso tem o mesmo efeito que MQZCI\_STOP

#### **MQZCI\_STOP**

Não continue com o próximo componente

### **CompCode**

Tipo: MQLONG - saída

Código de conclusão. Deve ser um dos valores a seguir:

#### **MQCC\_OK**

Indica conclusão bem-sucedida.

#### **MQCC\_WARNING**

Conclusão parcial.

#### **MQCC\_FAILED**

A chamada falhou.

### **Razão**

Tipo: MQLONG - saída

Código de razão qualificando *CompCode*.

Se *CompCode* for MQCC\_OK:

#### **MQRC\_NONE**

(0, X'000') Nenhuma razão a relatar.

Se *CompCode* for MQCC\_WARNING:

#### **MQRC\_CHAR\_ATTRS\_TOO\_SHORT**

Não há espaço suficiente para os atributos de caracteres

#### **MQRC\_INT\_COUNT\_TOO\_SMALL**

Não há espaço suficiente para atributos inteiros.

Se *CompCode* for MQCC\_FAILED:

#### **MQRC\_SELECTOR\_COUNT\_ERROR**

O número de seletores não é válido

#### **MQRC\_SELECTOR\_ERROR**

Seletor de atributo inválido.

#### **MQRC\_SELECTOR\_LIMIT\_EXCEDIDO**

Muitos seletores especificados.

#### **MQRC\_INT\_ATTR\_COUNT\_ERROR**

O número de atributos de número inteiro não é válido.

#### **MQRC\_INT\_ATTRS\_ARRAY\_ERROR**

Matriz de atributos de número inteiro inválida.

#### **MQRC\_CHAR\_ATTR\_LENGTH\_ERROR**

O número de atributos de caracteres não é válido

#### **MQRC\_CHAR\_ATTRS\_ERROR**

A cadeia de atributos de caracteres não é válida

#### **MQRC\_SERVICE\_ERROR**

(2289, X'8F1') Ocorreu um erro inesperado ao acessar o serviço.

Para obter mais informações sobre esses códigos de razão, consulte [conclusão da API e códigos de razão](#).

## Chamada C

```
MQZ_INQUIRE (QMgrName, SelectorCount, Selectors, IntAttrCount,  
             &IntAttrs, CharAttrLength, &CharAttrs,  
             SelectorReturned, ComponentData, &Continuation,  
             &CompCode, &Reason);
```

Os parâmetros transmitidos para o serviço são declarados da seguinte forma:

```
MQCHAR48  QMgrName;           /* Queue manager name */  
MQLONG    SelectorCount;      /* Selector count */  
MQLONG    Selectors[n];       /* Selectors */  
MQLONG    IntAttrCount;       /* IntAttrs count */  
MQLONG    IntAttrs[n];        /* Integer attributes */  
MQLONG    CharAttrCount;      /* CharAttrs count */  
MQLONG    CharAttrs[n];       /* Character attributes */  
MQLONG    SelectorReturned[n]; /* Selector returned */  
MQBYTE    ComponentData[n];   /* Component data */  
MQLONG    Continuation;       /* Continuation indicator set by  
                               component */  
MQLONG    CompCode;           /* Completion code */  
MQLONG    Reason;             /* Reason code qualifying CompCode */
```

## MQZ\_REFRESH\_CACHE-Atualizar todas as autorizações

Essa função é fornecida por um componente de serviço de autorização MQZAS\_VERSION\_3 e é chamada pelo gerenciador de filas para atualizar a lista de autorizações mantidas internamente pelo componente.

O identificador de função para essa função (para MQZEP) é MQZID\_REFRESH\_CACHE (8L).

### Sintaxe

```
MQZ_REFRESH_CACHE( QMgrName , ComponentData , Continuation , CompCode ,  
Reason )
```

### Parâmetros

#### QMgrName

Tipo: MQCHAR48 - entrada

Nome do gerenciador de filas. O nome do gerenciador de fila chamando o componente. Esse nome é preenchido com espaços em branco até o comprimento total do parâmetro; o nome não é terminado por um caractere nulo.

O nome do gerenciador de filas é transmitido para o componente para obter informações; a interface de serviço de autorização não requer que o componente faça uso dele de nenhuma maneira definida

#### ComponentData

Tipo: MQBYTE xComponentDataComprimento-entrada/saída

Dados do componente.. Esses dados são mantidos pelo gerenciador de filas em nome desse componente específico; quaisquer mudanças feitas nele por qualquer uma das funções fornecidas por esse componente são preservadas e apresentadas na próxima vez em que uma das funções desse componente for chamada de.

O comprimento dessa área de dados é passado pelo gerenciador de filas no parâmetro **ComponentDataLength** da chamada MQZ\_INIT\_AUTHORITY

#### Continuação

Tipo: MQLONG - saída

Indicador de continuação configurado por componente.. Os valores a seguir podem ser especificados:

#### MQZCI\_DEFAULT

Continuação dependente do gerenciador de filas..

Para MQZ\_CHECK\_AUTHORITY, isso tem o mesmo efeito que MQZCI\_STOP

## **MQZCI\_CONTINUE**

Continue com o próximo componente

## **MQZCI\_STOP**

Não continue com o próximo componente

### **CompCode**

Tipo: MQLONG - saída

Código de conclusão. Deve ser um dos valores a seguir:

### **MQCC\_OK**

Indica conclusão bem-sucedida.

### **MQCC\_FAILED**

A chamada falhou.

### **Razão**

Tipo: MQLONG - saída

Código de razão qualificando *CompCode*.

Se *CompCode* for MQCC\_OK:

### **MQRC\_NONE**

(0, X'000') Nenhuma razão a relatar.

Se *CompCode* for MQCC\_WARNING:

### **MQRC\_SERVICE\_ERROR**

(2289, X'8F1') Ocorreu um erro inesperado ao acessar o serviço.

## **Chamada C**

```
MQZ_REFRESH_CACHE (QMgrName, ComponentData,  
                  &Continuation, &CompCode, &Reason);
```

Declare os parâmetros conforme a seguir:

```
MQCHAR48  QMgrName;           /* Queue manager name */  
MQBYTE    ComponentData[n]; /* Component data */  
MQLONG    Continuation;     /* Continuation indicator set by  
                             component */  
MQLONG    CompCode;         /* Completion code */  
MQLONG    Reason;          /* Reason code qualifying CompCode */
```

## **MQZ\_SET\_AUTHORITY-Configurar autoridade**

Esta função é fornecida por um componente de serviço de autorização MQZAS\_VERSION\_1 e é iniciada pelo gerenciador de filas para configurar a autoridade que uma entidade tem para acessar o objeto especificado.

O identificador de função para esta função (para MQZEP) é MQZID\_SET\_AUTHORITY

**Nota:** Esta função substitui quaisquer autoridades existentes. Para preservar quaisquer autoridades existentes, você deve configurá-las novamente com essa função.

### **Sintaxe**

```
MQZ_SET_AUTHORITY( QMgrName , EntityName , EntityType , ObjectName ,  
ObjectType , Authority , ComponentData , Continuation , CompCode , Reason )
```

### **Parâmetros**

#### **QMgrName**

Tipo: MQCHAR48 - entrada



Nome do gerenciador de filas. O nome do gerenciador de fila chamando o componente. Esse nome é preenchido com espaços em branco até o comprimento total do parâmetro; o nome não é terminado por um caractere nulo.

O nome do gerenciador de filas é transmitido para o componente para obter informações; a interface de serviço de autorização não requer que o componente o use de nenhuma maneira definida

### **EntityName**

Tipo: MQCHAR12 - entrada

Nome da entidade.. O nome da entidade para a qual o acesso ao objeto será recuperado. O comprimento máximo da sequência é de 12 caracteres; se ela for mais curta do que ela for preenchida à direita com espaços em branco. O nome não é terminado por um caractere nulo.

### **EntityType**

Tipo: MQLONG - entrada

Tipo de entidade.. O tipo de entidade especificado por *EntityName* Deve ser um dos valores a seguir:

#### **MQZAET\_PRINCIPAL**

Principal.

#### **MQZAET\_GROUP**

### **ObjectName**

Tipo: MQCHAR48 - entrada

Nome do objeto. O nome do objeto para o qual o acesso é necessário O comprimento máximo da sequência é de 48 caracteres; se for mais curto do que isso, será preenchido à direita com espaços em branco. O nome não é terminado por um caractere nulo.

Se *ObjectType* for MQOT\_Q\_MGR, esse nome será o mesmo que *QMgrName*

### **ObjectType**

Tipo: MQLONG - entrada

Tipo de objeto. O tipo de entidade especificado por *ObjectName* Deve ser um dos valores a seguir:

#### **MQOT\_AUTH\_INFO**

Informações de autenticação..

#### **MQOT\_CHANNEL**

Canal.

#### **MQOT\_CLNTCONN\_CHANEXO L**

Canal de conexão do cliente.

#### **MQOT\_LISTENER**

Ouvinte.

#### **MQOT\_NAMELIST**

Lista de nomes

#### **MQOT\_PROCESS**

process definition.

#### **MQOT\_Q**

Fila.

#### **MQOT\_Q\_MGR**

Gerenciador de Filas

#### **MQOT\_SERVICE**

Serviço.

#### **MQOT\_TOPIC**

:NONE.

### **Autoridade**

Tipo: MQLONG - entrada

Autoridade da entidade.. Se uma autoridade estiver sendo configurada, esse campo será igual à operação de autorização apropriada (constante MQZAO\_\*) Se mais de uma autoridade estiver sendo configurada, este campo será a OR bit a bit das constantes MQZAO\_\* correspondentes.

#### **ComponentDatarname>**

Tipo: MQBYTEComponentDataLength -entrada/saída.

Dados do componente.. Esses dados são mantidos pelo gerenciador de filas em nome desse componente específico; quaisquer mudanças feitas nele por qualquer uma das funções fornecidas por esse componente são preservados e apresentados na próxima vez em que uma dessas funções do componente for chamada

O comprimento dessa área de dados é passado pelo gerenciador de filas no parâmetro

**ComponentDataLength** da chamada MQZ\_INIT\_AUTHORITY

#### **Continuação**

Tipo: MQLONG - saída

Indicador de continuação configurado por componente.. Os valores a seguir podem ser especificados:

##### **MQZCI\_DEFAULT**

Continuação dependente do gerenciador de filas..

Para MQZ\_GET\_AUTHORITY, isso tem o mesmo efeito que MQZCI\_CONTINUE

##### **MQZCI\_CONTINUE**

Continue com o próximo componente

##### **MQZCI\_STOP**

Não continue com o próximo componente

#### **CompCode**

Tipo: MQLONG - saída

Código de conclusão. Deve ser um dos valores a seguir:

##### **MQCC\_OK**

Indica conclusão bem-sucedida.

##### **MQCC\_FAILED**

A chamada falhou.

#### **Razão**

Tipo: MQLONG - saída

Código de razão qualificando *CompCode*.

Se *CompCode* for MQCC\_OK:

##### **MQRC\_NONE**

(0, X'000') Nenhuma razão a relatar.

Se *CompCode* for MQCC\_FAILED:

##### **MQRC\_NOT\_AUTHORIZED**

(2035, X'7F3') Não autorizado para acesso.

##### **MQRC\_SERVICE\_ERROR**

(2289, X'8F1') Ocorreu um erro inesperado ao acessar o serviço.

##### **MQRC\_SERVICE\_NOT\_AVAILABLE**

(2285, X'8ED') Serviço subjacente não disponível.

##### **MQRC\_UNKNOWN\_ENTITY**

(2292, X'8F4') Entidade desconhecida para serviço.

Para obter mais informações sobre esses códigos de razão, consulte [conclusão da API e códigos de razão](#).

## Chamada C

```
MQZ_SET_AUTHORITY (QMgrName, EntityName, EntityType, ObjectName,  
                  ObjectType, Authority, ComponentData,  
                  &Continuation, &CompCode, &Reason);
```

Os parâmetros transmitidos para o serviço são declarados da seguinte forma:

```
MQCHAR48  QMgrName;           /* Queue manager name */  
MQCHAR12  EntityName;        /* Entity name */  
MQLONG    EntityType;        /* Entity type */  
MQCHAR48  ObjectName;        /* Object name */  
MQLONG    ObjectType;        /* Object type */  
MQLONG    Authority;         /* Authority to be checked */  
MQBYTE    ComponentData[n]; /* Component data */  
MQLONG    Continuation;      /* Continuation indicator set by  
                             component */  
MQLONG    CompCode;          /* Completion code */  
MQLONG    Reason;           /* Reason code qualifying CompCode */
```

### MQZ\_SET\_AUTHORITY\_2 -Configurar autoridade (estendido)

Essa função é fornecida por um componente de serviço de autorização MQZAS\_VERSION\_2 e é iniciada pelo gerenciador de filas para configurar a autoridade que uma entidade tem para acessar o objeto especificado.

O identificador de função para esta função (para MQZEP) é MQZID\_SET\_AUTHORITY

**Nota:** Esta função substitui quaisquer autoridades existentes. Para preservar quaisquer autoridades existentes, você deve configurá-las novamente com essa função.

MQZ\_SET\_AUTHORITY\_2 é como MQZ\_SET\_AUTHORITY, mas com o parâmetro **EntityName** substituído pelo parâmetro **EntityData**.

### Sintaxe

```
MQZ_SET_AUTHORITY_2( QMgrName , EntityData , EntityType , ObjectName ,  
ObjectType , Authority , ComponentData , Continuation , CompCode , Reason )
```

### Parâmetros

#### QMgrName

Tipo: MQCHAR48 - entrada

Nome do gerenciador de filas. O nome do gerenciador de fila chamando o componente. Esse nome é preenchido com espaços em branco até o comprimento total do parâmetro; o nome não é terminado por um caractere nulo.

O nome do gerenciador de filas é transmitido para o componente para obter informações; a interface de serviço de autorização não requer que o componente o use de nenhuma maneira definida

#### EntityData

Tipo: MQZED-entrada

Dados da entidade.. Dados relacionados à entidade cuja autorização para o objeto deve ser configurada Consulte [“MQZED-Descrição de entidade” na página 1737](#) para obter detalhes.

#### EntityType

Tipo: MQLONG - entrada

Tipo de entidade.. O tipo de entidade especificado por *EntityData* Deve ser um dos valores a seguir:

#### MQZAET\_PRINCIPAL

Principal.

## **MQZAET\_GROUP**

### **ObjectName**

Tipo: MQCHAR48 - entrada

Nome do objeto. O nome do objeto para o qual a autoridade de entidade deve ser configurada O comprimento máximo da sequência é de 48 caracteres; se for mais curto do que isso, será preenchido à direita com espaços em branco. O nome não é terminado por um caractere nulo.

Se *ObjectType* for MQOT\_Q\_MGR, esse nome será o mesmo que *QMgrName*

### **ObjectType**

Tipo: MQLONG - entrada

Tipo de objeto. O tipo de entidade especificado por *ObjectName* Deve ser um dos valores a seguir:

#### **MQOT\_AUTH\_INFO**

Informações de autenticação..

#### **MQOT\_CHANNEL**

Canal.

#### **MQOT\_CLNTCONN\_CHANEXOL**

Canal de conexão do cliente.

#### **MQOT\_LISTENER**

Ouvinte.

#### **MQOT\_NAMELIST**

Lista de nomes

#### **MQOT\_PROCESS**

process definition.

#### **MQOT\_Q**

Fila.

#### **MQOT\_Q\_MGR**

Gerenciador de Filas

#### **MQOT\_SERVICE**

Serviço.

#### **MQOT\_TOPIC**

:NONE.

### **Autoridade**

Tipo: MQLONG - entrada

Autoridade da entidade.. Se uma autoridade estiver sendo configurada, esse campo será igual à operação de autorização apropriada (constante MQZAO\_\*) Se mais de uma autoridade estiver sendo configurada, este campo será a OR bit a bit das constantes MQZAO\_\* correspondentes.

### **ComponentData**

Tipo: MQBYTE xComponentDataComprimento-entrada/saída

Dados do componente.. Esses dados são mantidos pelo gerenciador de filas em nome desse componente específico; quaisquer mudanças feitas nele por qualquer uma das funções fornecidas por esse componente são preservados e apresentados na próxima vez em que uma dessas funções do componente for chamada

O comprimento dessa área de dados é passado pelo gerenciador de filas no parâmetro

**ComponentDataLength** da chamada MQZ\_INIT\_AUTHORITY

### **Continuação**

Tipo: MQLONG - saída

Indicador de continuação configurado por componente.. Os valores a seguir podem ser especificados:

**MQZCI\_DEFAULT**

Continuação dependente do gerenciador de filas..

Para MQZ\_CHECK\_AUTHORITY, isso tem o mesmo efeito que MQZCI\_STOP

**MQZCI\_CONTINUE**

Continue com o próximo componente

**MQZCI\_STOP**

Não continue com o próximo componente

**CompCode**

Tipo: MQLONG - saída

Código de conclusão. Deve ser um dos valores a seguir:

**MQCC\_OK**

Indica conclusão bem-sucedida.

**MQCC\_FAILED**

A chamada falhou.

**Razão**

Tipo: MQLONG - saída

Código de razão qualificando *CompCode*.

Se *CompCode* for MQCC\_OK:

**MQRC\_NONE**

(0, X'000') Nenhuma razão a relatar.

Se *CompCode* for MQCC\_FAILED:

**MQRC\_NOT\_AUTHORIZED**

(2035, X'7F3') Não autorizado para acesso.

**MQRC\_SERVICE\_ERROR**

(2289, X'8F1') Ocorreu um erro inesperado ao acessar o serviço.

**MQRC\_SERVICE\_NOT\_AVAILABLE**

(2285, X'8ED') Serviço subjacente não disponível.

**MQRC\_UNKNOWN\_ENTITY**

(2292, X'8F4') Entidade desconhecida para serviço.

Para obter mais informações sobre esses códigos de razão, consulte [conclusão da API e códigos de razão](#).

**Chamada C**

```
MQZ_SET_AUTHORITY_2 (QMgrName, &EntityData, EntityType, ObjectName,
                    ObjectType, Authority, ComponentData,
                    &Continuation, &CompCode, &Reason);
```

Os parâmetros transmitidos para o serviço são declarados da seguinte forma:

```
MQCHAR48  QMgrName;           /* Queue manager name */
MQZED     EntityData;        /* Entity data */
MQLONG    EntityType;        /* Entity type */
MQCHAR48  ObjectName;        /* Object name */
MQLONG    ObjectType;        /* Object type */
MQLONG    Authority;         /* Authority to be checked */
MQBYTE    ComponentData[n]; /* Component data */
MQLONG    Continuation;      /* Continuation indicator set by
                             component */
MQLONG    CompCode;          /* Completion code */
MQLONG    Reason;            /* Reason code qualifying CompCode */
```

## MQZ\_TERM\_AUTHORITY-Finalizar serviço de autorização

Essa função é fornecida por um componente de serviço de autorização e é iniciada pelo gerenciador de filas quando não requer mais os serviços desse componente. A função deve executar qualquer limpeza necessária pelo componente.

O identificador de função para essa função (para MQZEP) é MQZID\_TERM\_AUTHORITY

### Sintaxe

MQZ\_TERM\_AUTHORITY( Hconfig , Options , QMgrName , ComponentData , CompCode , Reason )

### Parâmetros

#### Hconfig

Tipo: MQHCONFIG-entrada

Identificador de configuração Esse identificador representa o componente específico que está sendo finalizado Ele deve ser usado pelo componente ao chamar o gerenciador de filas com a função MQZEP

#### Opções

Tipo: MQLONG - entrada

Opções de finalização Deve ser um dos valores a seguir:

#### **MQZTO\_PRIMARY**

Término primário

#### **MQZTO\_SECONDARY**

Finalização secundária

#### QMgrName

Tipo: MQCHAR48 - entrada

Nome do gerenciador de filas. O nome do gerenciador de fila chamando o componente. Esse nome é preenchido com espaços em branco até o comprimento total do parâmetro; o nome não é terminado por um caractere nulo.

O nome do gerenciador de filas é transmitido para o componente para obter informações; a interface de serviço de autorização não requer que o componente o use de nenhuma maneira definida

#### ComponentData

Tipo: MQBYTE x ComponentDataComprimento-entrada/saída

Dados do componente.. Esses dados são mantidos pelo gerenciador de filas em nome desse componente específico; quaisquer mudanças feitas nele por qualquer uma das funções fornecidas por esse componente são preservados e apresentados na próxima vez em que uma dessas funções do componente for chamada

O comprimento dessa área de dados é transmitido pelo gerenciador de filas no parâmetro de Comprimento ComponentData chamada MQZ\_INIT\_AUTHORITY

Quando a chamada MQZ\_TERM\_AUTHORITY tiver sido concluída, o gerenciador de filas descartará esses dados

#### CompCode

Tipo: MQLONG - saída

Código de conclusão. Deve ser um dos valores a seguir:

#### **MQCC\_OK**

Indica conclusão bem-sucedida.

#### **MQCC\_FAILED**

A chamada falhou.

## Razão

Tipo: MQLONG - saída

Código de razão qualificando *CompCode*.

Se *CompCode* for MQCC\_OK:

### **MQRC\_NONE**

(0, X'000') Nenhuma razão a relatar.

Se *CompCode* for MQCC\_FAILED:

### **MQRC\_SERVICE\_NOT\_AVAILABLE**

(2285, X'8ED') Serviço subjacente não disponível.

### **MQRC\_TERMINATION\_FAILED**

(2287, X'8FF') A finalização falhou por uma razão indefinida.

Para obter mais informações sobre esses códigos de razão, consulte [conclusão da API e códigos de razão](#).

## Chamada C

```
MQZ_TERM_AUTHORITY (Hconfig, Options, QMgrName, ComponentData,  
&CompCode, &Reason);
```

Os parâmetros transmitidos para o serviço são declarados da seguinte forma:

```
MQHCONFIG  Hconfig;           /* Configuration handle */  
MQLONG     Options;          /* Termination options */  
MQCHAR48   QMgrName;        /* Queue manager name */  
MQBYTE     ComponentData[n]; /* Component data */  
MQLONG     CompCode;        /* Completion code */  
MQLONG     Reason;          /* Reason code qualifying CompCode */
```

## MQZ\_DELETE\_NAME-Excluir nome

Essa função é fornecida por um componente de serviço de nomes e é iniciada pelo gerenciador de fila para excluir uma entrada para a fila especificada.

O identificador de função para esta função (para MQZEP) é MQZID\_DELETE\_NAME.

### Sintaxe

```
MQZ_DELETE_NAME( QMgrName , QName , ComponentData , Continuation , CompCode ,  
Reason )
```

### Parâmetros

#### **QMgrName**

Tipo: MQCHAR48 - entrada

Nome do gerenciador de filas. O nome do gerenciador de fila chamando o componente. Esse nome é preenchido com espaços em branco até o comprimento total do parâmetro; o nome não é terminado por um caractere nulo.

O nome do gerenciador de filas é transmitido para o componente para obter informações; a interface de serviço de autorização não requer que o componente o use de nenhuma maneira definida

#### **QName**

Tipo: MQCHAR48 - entrada

Nome da Fila. O nome da fila para a qual uma entrada será excluída. Esse nome é preenchido com espaços em branco até o comprimento total do parâmetro; o nome não é terminado por um caractere nulo.

## ComponentData

Tipo: MQBYTE x ComponentDataComprimento-entrada/saída

Dados do componente.. Esses dados são mantidos pelo gerenciador de filas em nome desse componente específico; quaisquer mudanças feitas nele por qualquer uma das funções fornecidas por esse componente são preservados e apresentados na próxima vez em que uma dessas funções do componente for chamada

O comprimento dessa área de dados é transmitido pelo gerenciador de filas no parâmetro de Comprimento ComponentData na chamada MQZ\_INIT\_NAME

## Continuação

Tipo: MQLONG - saída

Indicador de continuação configurado por componente.. Deve ser um dos valores a seguir:

### MQZCI\_DEFAULT

Continuação dependente do gerenciador de filas..

### MQZCI\_STOP

Não continue com o próximo componente

Para o comando **MQZ\_DELETE\_NAME**, o gerenciador de filas não tenta iniciar outro componente, não importa o que é retornado no parâmetro **Continuation**

## CompCode

Tipo: MQLONG - saída

Código de conclusão. Deve ser um dos valores a seguir:

### MQCC\_OK

Indica conclusão bem-sucedida.

### MQCC\_WARNING

Aviso (conclusão parcial).

### MQCC\_FAILED

A chamada falhou.

## Razão

Tipo: MQLONG - saída

Código de razão qualificando *CompCode*.

Se *CompCode* for MQCC\_OK:

### MQRC\_NONE

(0, X'000') Nenhuma razão a relatar.

Se *CompCode* for MQCC\_WARNING:

### MQRC\_UNKNOWN\_NAME

(2288, X'8F0') Nome da fila não localizado.

**Nota:** Talvez não seja possível retornar esse código, se o serviço subjacente responder com sucesso para esse caso

Se *CompCode* for MQCC\_FAILED:

### MQRC\_SERVICE\_ERROR

(2289, X'8F1') Ocorreu um erro inesperado ao acessar o serviço.

### MQRC\_SERVICE\_NOT\_AVAILABLE

(2285, X'8ED') Serviço subjacente não disponível.

Para obter mais informações sobre esses códigos de razão, consulte [conclusão da API e códigos de razão](#).



## Chamada C

```
MQZ_DELETE_NAME (QMgrName, QName, ComponentData, &Continuation,  
&CompCode, &Reason);
```

Os parâmetros transmitidos para o serviço são declarados da seguinte forma:

```
MQCHAR48  QMgrName;           /* Queue manager name */  
MQCHAR48  QName;             /* Queue name */  
MQBYTE    ComponentData[n]; /* Component data */  
MQLONG    Continuation;      /* Continuation indicator set by  
                             component */  
MQLONG    CompCode;          /* Completion code */  
MQLONG    Reason;           /* Reason code qualifying CompCode */
```

## MQZ\_INIT\_NAME-Inicializar serviço de nomes

Essa função é fornecida por um componente de serviço de nomes e é iniciada pelo gerenciador de fila durante a configuração do componente... É esperado chamar MQZEP para fornecer informações para o gerenciador de filas.

O identificador de função para essa função (para MQZEP) é MQZID\_INIT\_NAME.

### Sintaxe

```
MQZ_INIT_NAME( Hconfig , Options , QMgrName , ComponentDataLength ,  
ComponentData , Version , CompCode , Reason )
```

### Parâmetros

#### Hconfig

Tipo: MQHCONFIG-entrada

Identificador de configuração Essa manipulação representa o componente específico sendo inicializado. Ele deve ser usado pelo componente ao chamar o gerenciador de filas com a função MQZEP

#### Opções

Tipo: MQLONG - entrada

Opções de inicialização.. Deve ser um dos valores a seguir:

#### **MQZIO\_PRIMARY**

Inicialização primária..

#### **MQZIO\_SECONDARY**

Inicialização secundária..

#### QMgrName

Tipo: MQCHAR48 - entrada

Nome do gerenciador de filas. O nome do gerenciador de fila chamando o componente. Esse nome é preenchido com espaços em branco até o comprimento total do parâmetro; o nome não é terminado por um caractere nulo.

O nome do gerenciador de filas é transmitido para o componente para obter informações; a interface de serviço de autorização não requer que o componente o use de nenhuma maneira definida

#### Comprimento de ComponentData

Tipo: MQLONG - entrada

Comprimento dos dados do componente Comprimento em bytes da área *ComponentData* . Esse comprimento é definido no componente de dados de configuração.

## ComponentData

Tipo: MQBYTE x ComponentDataComprimento-entrada/saída

Dados do componente.. É inicializado para todos os zeros antes de chamar a função de inicialização primária do componente. Esses dados são mantidos pelo gerenciador de filas em nome desse componente específico; quaisquer mudanças feitas nele por qualquer uma das funções (incluindo a função de inicialização) fornecidas por esse componente são preservadas e apresentadas na próxima vez em que uma dessas funções do componente for chamada.

O comprimento dessa área de dados é passado pelo gerenciador de filas no parâmetro **ComponentDataLength** da chamada MQZ\_INIT\_AUTHORITY

## Versão

Tipo: MQLONG-entrada/saída

Número da versão.. Na entrada para a função de inicialização, isso identifica o número da versão mais alta que o gerenciador de fila suporta.. A função de inicialização deve alterar isso, se necessário, para a versão da interface que ela suporta.. Se no retorno o gerenciador de filas não suportar a versão retornada pelo componente, ele chamará a função MQZ\_TERM\_NAME do componente e não fará uso adicional desse componente.

Os seguintes valores são suportados:

### **MQZAS\_VERSION\_1**

Versão 1.

## CompCode

Tipo: MQLONG - saída

Código de conclusão. Deve ser um dos valores a seguir:

### **MQCC\_OK**

Indica conclusão bem-sucedida.

### **MQCC\_FAILED**

A chamada falhou.

## Razão

Tipo: MQLONG - saída

Código de razão qualificando *CompCode*.

Se *CompCode* for MQCC\_OK:

### **MQRC\_NONE**

(0, X'000') Nenhuma razão a relatar.

Se *CompCode* for MQCC\_FAILED:

### **MQRC\_INITIALIZATION\_FAILED**

(2286, X'8EE') A inicialização falhou por uma razão indefinida..

### **MQRC\_SERVICE\_NOT\_AVAILABLE**

(2285, X'8ED') Serviço subjacente não disponível.

Para obter mais informações sobre esses códigos de razão, consulte [conclusão da API e códigos de razão](#).

## Chamada C

```
MQZ_INIT_NAME (Hconfig, Options, QMgrName, ComponentDataLength,  
              ComponentData, &Version, &CompCode, &Reason);
```

Os parâmetros transmitidos para o serviço são declarados da seguinte forma:

```
MQHCONFIG  Hconfig;          /* Configuration handle */  
MQLONG     Options;         /* Initialization options */
```

```

MQCHAR48  QMgrName;           /* Queue manager name */
MQLONG    ComponentDataLength; /* Length of component data */
MQBYTE    ComponentData[n];  /* Component data */
MQLONG    Version;          /* Version number */
MQLONG    CompCode;         /* Completion code */
MQLONG    Reason;           /* Reason code qualifying CompCode */

```

## MQZ\_INSERT\_NAME-Inserir nome

Essa função é fornecida por um componente de serviço de nomes e é iniciada pelo gerenciador de filas para inserir uma entrada para a fila especificada, contendo o nome do gerenciador de filas que possui a fila. Se a fila já estiver definida no serviço, a chamada falha.

O identificador de função para essa função (para MQZEP) é MQZID\_INSERT\_NAME.

### Sintaxe

```

MQZ_INSERT_NAME( QMgrName , QName , ResolvedQMgrName , ComponentData ,
Continuation , CompCode , Reason )

```

### Parâmetros

#### QMgrName

Tipo: MQCHAR48 - entrada

Nome do gerenciador de filas. O nome do gerenciador de fila chamando o componente. Esse nome é preenchido com espaços em branco até o comprimento total do parâmetro; o nome não é terminado por um caractere nulo.

O nome do gerenciador de filas é transmitido para o componente para obter informações; a interface de serviço de autorização não requer que o componente o use de nenhuma maneira definida

#### QName

Tipo: MQCHAR48 - entrada

Nome da Fila. O nome da fila para a qual uma entrada deve ser inserida. Esse nome é preenchido com espaços em branco até o comprimento total do parâmetro; o nome não é terminado por um caractere nulo.

#### ResolvedQMgrName

Tipo: MQCHAR48 - entrada

Nome do gerenciador de filas resolvido. O nome do gerenciador de fila para o qual a fila é resolvida. Esse nome é preenchido com espaços em branco até o comprimento total do parâmetro; o nome não é terminado por um caractere nulo.

#### ComponentData

Tipo: MQBYTE xComponentDataComprimento-entrada/saída

Dados do componente.. Esses dados são mantidos pelo gerenciador de filas em nome desse componente específico; quaisquer mudanças feitas nele por qualquer uma das funções (incluindo a função de inicialização) fornecidas por esse componente são preservadas e apresentadas na próxima vez em que uma dessas funções do componente for chamada.

O comprimento dessa área de dados é transmitido pelo gerenciador de filas no parâmetro **ComponentDataLength** da chamada MQZ\_INIT\_NAME

#### Continuação

Tipo: MQLONG-entrada/saída

Indicador de continuação configurado por componente.. Para MQZ\_INSERT\_NAME, o gerenciador de filas não tenta iniciar outro componente, o que for retornado no parâmetro **Continuation**.

Os seguintes valores são suportados:

#### MQZCI\_DEFAULT

Continuação dependente do gerenciador de filas..

## **MQZCI\_STOP**

Não continue com o próximo componente

### **CompCode**

Tipo: MQLONG - saída

Código de conclusão. Deve ser um dos valores a seguir:

#### **MQCC\_OK**

Indica conclusão bem-sucedida.

#### **MQCC\_FAILED**

A chamada falhou.

### **Razão**

Tipo: MQLONG - saída

Código de razão qualificando *CompCode*.

Se *CompCode* for MQCC\_OK:

#### **MQRC\_NONE**

(0, X'000') Nenhuma razão a relatar.

Se *CompCode* for MQCC\_FAILED:

#### **MQRC\_Q\_ALREADY\_EXISTS**

(2290, X'8F2') O objeto Queue já existe.

#### **MQRC\_SERVICE\_ERROR**

(2289, X'8F1') Ocorreu um erro inesperado ao acessar o serviço.

#### **MQRC\_SERVICE\_NOT\_AVAILABLE**

(2285, X'8ED') Serviço subjacente não disponível.

Para obter mais informações sobre esses códigos de razão, consulte [conclusão da API e códigos de razão](#).

## **Chamada C**

```
MQZ_INSERT_NAME (QMgrName, QName, ResolvedQMgrName, ComponentData,  
&Continuation, &CompCode, &Reason);
```

Os parâmetros transmitidos para o serviço são declarados da seguinte forma:

```
MQCHAR48 QMgrName;          /* Queue manager name */  
MQCHAR48 QName;             /* Queue name */  
MQCHAR48 ResolvedQMgrName;  /* Resolved queue manager name */  
MQBYTE ComponentData[n];    /* Component data */  
MQLONG Continuation;        /* Continuation indicator set by  
                             component */  
MQLONG CompCode;            /* Completion code */  
MQLONG Reason;              /* Reason code qualifying CompCode */
```

## **MQZ\_LOOKUP\_NAME-Nome da consulta**

Essa função é fornecida por um componente de serviço de nomes e é iniciada por um gerenciador de filas para recuperar o nome do gerenciador de filas proprietário para uma fila especificada.

O identificador de função para essa função (para MQZEP) é MQZID\_LOOKUP\_NAME

### **Sintaxe**

```
MQZ_LOOKUP_NAME( QMgrName , QName , ResolvedQMgrName , ComponentData ,  
Continuation , CompCode , Reason )
```

## Parâmetros

### QMgrName

Tipo: MQCHAR48 - entrada

Nome do gerenciador de filas. O nome do gerenciador de fila chamando o componente. Esse nome é preenchido com espaços em branco até o comprimento total do parâmetro; o nome não é terminado por um caractere nulo.

O nome do gerenciador de filas é transmitido para o componente para obter informações; a interface de serviço de autorização não requer que o componente o use de nenhuma maneira definida

### QName

Tipo: MQCHAR48 - entrada

Nome da Fila. O nome da fila para a qual uma entrada deve ser resolvida. Esse nome é preenchido com espaços em branco até o comprimento total do parâmetro; o nome não é terminado por um caractere nulo.

### ResolvedQMgrName

Tipo: MQCHAR48 -saída

Nome do gerenciador de filas resolvido. Se a função for concluída com êxito, este será o nome do gerenciador de filas que possui a fila

O nome retornado pelo componente de serviço deve ser preenchido à direita com espaços em branco até o comprimento total do parâmetro; o nome não deve ser terminado por um caractere nulo ou conter espaços em branco iniciais ou integrados.

### ComponentData

Tipo: MQBYTEXComponentDataLength -entrada/saída.

Dados do componente.. Esses dados são mantidos pelo gerenciador de filas em nome desse componente específico; quaisquer mudanças feitas nele por qualquer uma das funções (incluindo a função de inicialização) fornecidas por esse componente são preservadas e apresentadas na próxima vez em que uma dessas funções do componente for chamada.

O comprimento dessa área de dados é transmitido pelo gerenciador de filas no parâmetro **ComponentDataLength** da chamada MQZ\_INIT\_NAME

### Continuação

Tipo: MQLONG - saída

Indicador de continuação configurado por componente.. Para MQZ\_LOOKUP\_NAME, o gerenciador de filas especifica se deve iniciar outro componente de serviço de nomes, conforme a seguir:

- Se *CompCode* for MQCC\_OK, nenhum componente adicional será iniciado, independentemente do valor retornado em *Continuação*.
- Se *CompCode* não for MQCC\_OK, um componente adicional será iniciado, a menos que *Continuation* seja MQZCI\_STOP.

Os seguintes valores são suportados:

#### **MQZCI\_DEFAULT**

Continuação dependente do gerenciador de filas..

#### **MQZCI\_CONTINUE**

Continue com o próximo componente

#### **MQZCI\_STOP**

Não continue com o próximo componente

### CompCode

Tipo: MQLONG - saída

Código de conclusão. Deve ser um dos valores a seguir:

**MQCC\_OK**

Indica conclusão bem-sucedida.

**MQCC\_FAILED**

A chamada falhou.

**Razão**

Tipo: MQLONG - saída

Código de razão qualificando *CompCode*.

Se *CompCode* for MQCC\_OK:

**MQRC\_NONE**

(0, X'000') Nenhuma razão a relatar.

Se *CompCode* for MQCC\_FAILED:

**MQRC\_SERVICE\_ERROR**

(2289, X'8F1') Ocorreu um erro inesperado ao acessar o serviço.

**MQRC\_SERVICE\_NOT\_AVAILABLE**

(2285, X'8ED') Serviço subjacente não disponível.

**MQRC\_UNKNOWN\_Q\_NAME**

(2288, X'8F0') Nome da fila não localizado.

Para obter mais informações sobre esses códigos de razão, consulte [conclusão da API e códigos de razão](#).

**Chamada C**

```
MQZ_LOOKUP_NAME (QMgrName, QName, ResolvedQMgrName, ComponentData,
&Continuation, &CompCode, &Reason);
```

Os parâmetros transmitidos para o serviço são declarados da seguinte forma:

```
MQCHAR48 QMgrName;          /* Queue manager name */
MQCHAR48 QName;             /* Queue name */
MQCHAR48 ResolvedQMgrName;  /* Resolved queue manager name */
MQBYTE ComponentData[n];   /* Component data */
MQLONG Continuation;       /* Continuation indicator set by
                             component */
MQLONG CompCode;           /* Completion code */
MQLONG Reason;             /* Reason code qualifying CompCode */
```

**MQZ\_TERM\_NAME-Finalizar serviço de nome**

Essa função é fornecida por um componente de serviço de nomes e é iniciada pelo gerenciador de fila quando não requer mais os serviços desse componente. A função deve executar qualquer limpeza necessária pelo componente.

O identificador de função para essa função (para MQZEP) é MQZID\_TERM\_NAME.

**Sintaxe**

```
MQZ_TERM_NAME( Hconfig , Options , QMgrName , ComponentData , CompCode ,
Reason )
```

**Parâmetros****Hconfig**

Tipo: MQHCONFIG-entrada

Identificador de configuração Esse identificador representa o componente específico que está sendo finalizado Ele é usado pelo componente ao chamar o gerenciador de fila com a função MQZEP

## Opções

Tipo: MQLONG - entrada

Opções de finalização Deve ser um dos valores a seguir:

### **MQZTO\_PRIMARY**

Término primário

### **MQZTO\_SECONDARY**

Finalização secundária

## QMgrName

Tipo: MQCHAR48 - entrada

Nome do gerenciador de filas. O nome do gerenciador de fila chamando o componente. Esse nome é preenchido com espaços em branco até o comprimento total do parâmetro; o nome não é terminado por um caractere nulo.

O nome do gerenciador de filas é transmitido para o componente para obter informações; a interface de serviço de autorização não requer que o componente o use de nenhuma maneira definida

## ComponentData

Tipo: MQBYTE x ComponentDataComprimento-entrada/saída

Dados do componente.. Esses dados são mantidos pelo gerenciador de filas em nome desse componente específico; quaisquer mudanças feitas nele por qualquer uma das funções (incluindo a função de inicialização) fornecidas por esse componente são preservadas e apresentadas na próxima vez em que uma dessas funções do componente for chamada.

Os dados do componente estão na memória compartilhada acessível a todos os processos

O comprimento dessa área de dados é transmitido pelo gerenciador de filas no parâmetro

**ComponentDataLength** da chamada MQZ\_INIT\_NAME

Quando a chamada MQZ\_TERM\_NAME tiver sido concluída, o gerenciador de filas descartará esses dados

## CompCode

Tipo: MQLONG - saída

Código de conclusão. Deve ser um dos valores a seguir:

### **MQCC\_OK**

Indica conclusão bem-sucedida.

### **MQCC\_FAILED**

A chamada falhou.

## Razão

Tipo: MQLONG - saída

Código de razão qualificando *CompCode*.

Se *CompCode* for MQCC\_OK:

### **MQRC\_NONE**

(0, X'000') Nenhuma razão a relatar.

Se *CompCode* for MQCC\_FAILED:

### **MQRC\_TERMINATION\_FAILED**

(2287, X'8FF') A finalização falhou por uma razão indefinida.

### **MQRC\_SERVICE\_NOT\_AVAILABLE**

(2285, X'8ED') Serviço subjacente não disponível.

Para obter mais informações sobre esses códigos de razão, consulte [conclusão da API e códigos de razão](#).

## Chamada C

```
MQZ_TERM_NAME (Hconfig, Options, QMgrName, ComponentData, &CompCode,  
&Reason);
```

Os parâmetros transmitidos para o serviço são declarados da seguinte forma:

```
MQHCONFIG Hconfig; /* Configuration handle */  
MQLONG Options; /* Termination options */  
MQCHAR48 QMgrName; /* Queue manager name */  
MQBYTE ComponentData[n]; /* Component data */  
MQLONG CompCode; /* Completion code */  
MQLONG Reason; /* Reason code qualifying CompCode */
```

## MQZAC-Contexto do aplicativo..

A estrutura MQZAC é usada na chamada MQZ\_AUTHENTICATE\_USER do parâmetro *ApplicationContext*. Esse parâmetro especifica os dados relacionados ao aplicativo de chamada

*Tabela 1* resume os campos na estrutura.

<i>Tabela 838. Campos em MQZAC.</i>	
<b>Campo</b>	<b>Descrição</b>
<u>StrucId</u>	Identificador de estruturação
<u>Versão</u>	Número de versão da estrutura
<u>ProcessId</u>	Identificador de Processo
<u>ThreadId</u>	Identificador de encadeamento
<u>ApplName</u>	Nome do aplicativo
<u>UserID</u>	Identificador de usuários
<u>EffectiveUser</u>	Identificador de usuário efetivo
<u>Ambiente</u>	Meio ambiente
<u>CallerType</u>	Tipo do Responsável pela Chamada
<u>AuthenticationType</u>	Tipo de autenticação
<u>BindType</u>	Tipo de ligação

## Campos

### StrucId

Tipo: MQCHAR4 -entrada

Identificador de estruturação. O valor é o seguinte:

#### **MQZAC\_STRUC\_ID**

Identificador para a estrutura de contexto do aplicativo.

Para a linguagem de programação C, a constante MQZAC\_STRUC\_ID\_ARRAY também é definida; isso tem o mesmo valor que MQZAC\_STRUC\_ID, mas é uma matriz de caracteres em vez de uma sequência.

### Versão

Tipo: MQLONG - entrada

Número de versão da estrutura. O valor é o seguinte:



**MQZAC\_VERSION\_1**

Estrutura do contexto de aplicativos Version-1 A constante MQZAC\_CURRENT\_VERSION especifica o número da versão atual.

**ProcessId**

Tipo: MQPID-entrada

Identificador de processo do aplicativo.

**ThreadId**

Tipo: MQTID-entrada

Identificador de encadeamento do aplicativo.

**ApplName**

Tipo: MQCHAR28 -entrada

Nome do aplicativo.

**UserID**

Tipo: MQCHAR12 -entrada

Identificador do usuário. No AIX and Linux esse campo especifica o ID do usuário real do aplicativo. Em Windows esse campo especifica o ID do usuário do aplicativo.

**ID do EffectiveUser**

Tipo: MQCHAR12 -entrada

Identificador de usuário efetivo Em AIX and Linux este campo especifica o ID do usuário efetivo do aplicativo. No Windows , esse campo está em branco

**Meio ambiente**

Tipo: MQLONG - entrada

Ambiente. Este campo especifica o ambiente a partir do qual a chamada foi feita O campo é um dos seguintes valores:

**MQXE\_COMMAND\_SERVER**

Servidor de Comandos

**MQXE\_MQSC**

Interpretador do comando **runmqsc** . de

**MQXE\_MCA**

Agente do canal de mensagens MQXE\_OTHER

**MQXE\_OTHER**

Ambiente indefinido

**CallerType**

Tipo: MQLONG - entrada

Tipo de Responsável pela Chamada Este campo especifica o tipo de programa que fez a chamada.. O campo é um dos seguintes valores:

**MQXACT\_EXTERNAL**

A chamada é externa ao gerenciador de filas.

**MQXACT\_INTERNAL**

A chamada é interna para o gerenciador de filas

**AuthenticationType**

Tipo: MQLONG - entrada

Tipo de autenticação. Este campo especifica o tipo de autenticação que está sendo executada O campo é um dos seguintes valores:

**MQZAT\_INITIAL\_CONTEXT**

A chamada de autenticação é devido ao contexto do usuário ser inicializado. Esse valor é usado durante uma chamada MQCONN ou MQCONNX.

## MQZAT\_CHANGE\_CONTEXT

A chamada de autenticação é devido ao contexto do usuário ser alterado. Esse valor é usado quando o MCA muda o contexto do usuário.. Tópico pai: MQZAC-

### BindType

Tipo: MQLONG - entrada

Tipo de ligação. Este campo especifica o tipo de ligação em uso. O campo é um dos seguintes valores:

#### MQCNO\_FASTPATH\_BINDING

Ligação de atalho.

#### MQCNO\_SHARED\_BINDING

Ligação compartilhada

#### MQCNO\_ISOLATED\_BINDING

Ligação isolada..

## Declaração C

Declare os campos da estrutura da seguinte forma:

```
typedef struct tagMQZAC MQZAC;
struct tagMQZAC {
    MQCHAR4   StrucId;           /* Structure identifier */
    MQLONG    Version;          /* Structure version number */
    MQPID     ProcessId;        /* Process identifier */
    MQTID     ThreadId;         /* Thread identifier */
    MQCHAR28  ApplName;         /* Application name */
    MQCHAR12  UserID;           /* User identifier */
    MQCHAR12  EffectiveUserID;  /* Effective user identifier */
    MQLONG    Environment;      /* Environment */
    MQLONG    CallerType;       /* Caller type */
    MQLONG    AuthenticationType; /* Authentication type */
    MQLONG    BindType;         /* Bind type */
};
```

## MQZAD-Dados de autoridade.

A estrutura MQZAD é usada na chamada MQZ\_ENUMERATE\_AUTHORITY\_DATA para dois parâmetros, uma entrada e uma saída.

Consulte “MQZ\_ENUMERATE\_AUTHORITY\_DATA-Enumerar dados de autoridade” na página 1695 para obter informações adicionais sobre os parâmetros **Filter** e **AuthorityBuffer** :

- MQZAD é usado para o parâmetro **Filter** que é entrada para a chamada. Este parâmetro especifica os critérios de seleção que devem ser usados para selecionar os dados de autoridade retornados pela chamada
- MQZAD também é usado para o parâmetro **AuthorityBuffer** que é saída da chamada. Este parâmetro especifica as autorizações para uma combinação de nome do perfil, tipo do objeto e entidade

*Tabela 1.* resume os campos na estrutura.

<i>Tabela 839. Campos em MQZAD</i>	
<b>Campo</b>	<b>Descrição</b>
<u>StrucId</u>	Identificador de estruturação
<u>Versão</u>	Número de versão da estrutura
<u>ProfileName</u>	Nome do Perfil
<u>ObjectType</u>	Tipo de objeto
<u>Autoridade</u>	Autoridade

Tabela 839. Campos em MQZAD (continuação)

<b>Campo</b>	<b>Descrição</b>
<u>EntityDataPtr</u>	Ponteiro para dados da entidade
<u>EntityType</u>	Tipo de entidade
<u>Opções</u>	Opções

## Campos

### StrucId

Tipo: MQCHAR4 - entrada

Identificador de estruturação. O valor é o seguinte:

#### **MQZAD\_STRUC\_ID**

Identificador para a estrutura de dados de autoridade

Para a linguagem de programação C, a constante MQZAD\_STRUC\_ID\_ARRAY também é definida; ela possui o mesmo valor que MQZAD\_STRUC\_ID, mas é uma matriz de caracteres em vez de uma sequência.

### Versão

Tipo: MQLONG - entrada

Número de versão da estrutura. O valor é o seguinte:

#### **MQZAD\_VERSION\_1**

Estrutura do contexto de aplicativos Version-1 A constante MQZAD\_CURRENT\_VERSION especifica o número da versão atual.

A constante a seguir especifica o número da versão atual:

#### **MQZAD\_CURRENT\_VERSION**

A versão atual da estrutura de dados de autoridade

### ProfileName

Tipo: MQCHAR48 - entrada

Nome do perfil.

Para o parâmetro **Filter**, este campo é o nome do perfil para o qual dados de autoridade são necessários. Se o nome estiver inteiramente em branco até o final do campo ou o primeiro caractere nulo, serão retornados dados de autoridade para todos os nomes de perfis.

Para o parâmetro **AuthorityBuffer**, esse campo é o nome de um perfil que corresponde aos critérios de seleção especificados

### ObjectType

Tipo: MQLONG - entrada

Tipo de objeto.

Para o parâmetro **Filter**, esse campo é o tipo de objeto para o qual os dados de autoridade são necessários. Se o valor for MQOT\_ALL, dados de autoridade para todos os tipos de objeto serão retornados.

Para o parâmetro **AuthorityBuffer**, esse campo é o tipo de objeto ao qual o perfil identificado pelo parâmetro **ProfileName** se aplica..

O valor é um dos seguintes; para o parâmetro **Filter**, o valor MQOT\_ALL também é válido:

#### **MQOT\_AUTH\_INFO**

Informações sobre Autenticação

#### **MQOT\_CHANNEL**

Canal

**MQOT\_CLNTCONN\_CHANEXOL**

Canal de conexão do cliente

**MQOT\_LISTENER**

Listener

**MQOT\_NAMELIST**

Lista de Nomes

**MQOT\_PROCESS**

Definição de processo

**MQOT\_Q**

Fila

**MQOT\_Q\_MGR**

Gerenciador de filas

**MQOT\_SERVICE**

Serviço

**Autoridade**

Tipo: MQLONG - entrada

Autoridade.

Para o parâmetro **Filter**, esse campo é ignorado..

Para o parâmetro **AuthorityBuffer**, esse campo representa as autorizações que a entidade tem para os objetos identificados por **ProfileName** e **ObjectType**. Se a entidade tiver apenas uma autoridade, o campo será igual ao valor de autorização apropriado (constante MQZAO\_\*). Se a entidade tiver mais de uma autoridade, o campo será OR bit a bit das constantes MQZAO\_\* correspondentes.

**EntityDataPtr**

Tipo: PMQZED-entrada

O endereço da estrutura MQZED que identifica uma entidade

Para o parâmetro **Filter**, esse campo aponta para uma estrutura MQZED que identifica a entidade para a qual dados de autoridade são necessários. Se **EntityDataPtr** for o ponteiro nulo, dados de autoridade para todas as entidades serão retornados.

Para o parâmetro **AuthorityBuffer**, esse campo aponta para uma estrutura MQZED que identifica a entidade para a qual dados de autoridade foram retornados.

**EntityType**

Tipo: MQLONG - entrada

Tipo de entidade..

Para o parâmetro **Filter**, esse campo especifica o tipo de entidade para o qual os dados de autoridade são necessários. Se o valor for MQZAET\_NONE, dados de autoridade para todos os tipos de entidade serão retornados.

Para o parâmetro **AuthorityBuffer**, esse campo especifica o tipo da entidade identificada na estrutura MQZED apontada pelo parâmetro **EntityDataPtr**.

O valor é um dos seguintes; para o parâmetro **Filter**, o valor MQZAET\_NONE também é válido:

**MQZAET\_PRINCIPAL**

Diretor

**MQZAET\_GROUP**

Group

**Opções**

Tipo: MQAUTHOPT-entrada

Opções. Este campo especifica opções que fornecem controle sobre os perfis exibidos. Um dos seguintes valores deve ser especificado:

**MQAUTHOPT\_NAME\_ALL\_MATCHING**

Exibe todos os perfis

**MQAUTHOPT\_NAME\_EXPLICIT**

Exibe perfis que possuem exatamente o mesmo nome especificado no campo **ProfileName** .

Além disso, um dos seguintes também deve ser especificado:

**MQAUTHOPT\_ENTITY\_SET**

Exiba todos os perfis que são usados para calcular a autoridade acumulativa que a entidade tem para o objeto especificado pelo parâmetro **ProfileName** O parâmetro **ProfileName** não deve conter nenhum caractere curinga.

- Se a entidade especificada for um principal, para cada membro do conjunto {entity, groups} o perfil mais aplicável que se aplica ao objeto será exibido.
- Se a entidade especificada for um grupo, o perfil mais aplicável do grupo que se aplica ao objeto será exibido.
- Se esse valor for especificado, então os valores de **ProfileName**, **ObjectType**, **EntityType** e o nome da entidade especificado na estrutura MQZED **EntityDataPtr** devem ser todos não em branco.

Se você tiver especificado MQAUTHOPT\_NAME\_ALL\_MATCHING, também será possível especificar o valor a seguir:

**MQAUTHOPT\_ENTITY\_EXPLICIT**

Exibe perfis que possuem exatamente o mesmo nome da entidade que o nome da entidade especificado na estrutura MQZED do **EntityDataPtr**

**Declaração C**

```
typedef struct tagMQZAD MQZAD;
struct tagMQZAD {
    MQCHAR4   StrucId;           /* Structure identifier */
    MQLONG    Version;          /* Structure version number */
    MQCHAR48  ProfileName;      /* Profile name */
    MQLONG    ObjectType;       /* Object type */
    MQLONG    Authority;        /* Authority */
    PMQZED    EntityDataPtr;    /* Address of MQZED structure identifying an
    entity */
    MQLONG    EntityType;       /* Entity type */
    MQAUTHOPT Options;         /* Options */
};
```

**MQZED-Descrição de entidade**

A estrutura MQZED é usada em várias chamadas de serviço de autorização para especificar a entidade para a qual a autorização deve ser verificada.

Tabela 1. resume os campos na estrutura.

Tabela 840. Campos em MQZED	
Campo	Descrição
StrucId	Identificador de estruturação
Versão	Versão
EntityName Ptr	Nome da entidade
EntityDomainPtr	Ponteiro do domínio de entidade
SecurityId	Identificador de segurança

Tabela 840. Campos em MQZED (continuação)

Campo	Descrição
<u>CorrelationPtr</u>	Ponteiro de correlação

## Campos

### StrucId

Tipo: MQCHAR4 -entrada

Identificador de estruturação. O valor é o seguinte:

#### **MQZED\_STRUC\_ID**

Identificador para a estrutura do descritor de entidade

Para a linguagem de programação C, a constante MQZED\_STRUC\_ID\_ARRAY também é definida; ela tem o mesmo valor que MQZED\_STRUC\_ID, mas é uma matriz de caracteres em vez de uma sequência.

### Versão

Tipo: MQLONG - entrada

Número de versão da estrutura. O valor é o seguinte:

#### **MQZED\_VERSION\_1**

Version-1 estrutura do descritor de entidade.

A constante a seguir especifica o número da versão atual:

#### **MQZED\_CURRENT\_VERSION**

A versão atual da estrutura do descritor de entidade

### EntityNamePtr

Tipo: PMQCHAR-entrada

Nome do perfil.

Endereço do nome da entidade.. Este é um ponteiro para o nome da entidade cuja autorização deve ser verificada

### EntityDomainPtr

Tipo: PMQCHAR-entrada

Endereço do nome de domínio da entidade Este é um ponteiro para o nome do domínio que contém a definição da entidade cuja autorização deve ser verificada

### SecurityId

Tipo: MQBYTE40 -entrada

Autoridade.

Identificador de segurança. Este é o identificador de segurança cuja autorização deve ser verificada

### CorrelationPtr

Tipo: MQPTR-entrada

Indicador de correlação. Isso facilita a transmissão de dados correlacionais entre a função autenticar usuário e outras funções OAM apropriadas.

## Declaração C

```
typedef struct tagMQZED MQZED;
struct tagMQZED {
    MQCHAR4    StrucId;           /* Structure identifier */
    MQLONG     Version;          /* Structure version number */
    PMQCHAR    EntityNamePtr;    /* Address of entity name */
    PMQCHAR    EntityDomainPtr;  /* Address of entity domain name */
}
```

```
MQBYTE40 SecurityId; /* Security identifier */
MQPTR CorrelationPtr; /* Address of correlation data */
```

## MQZEP-Incluir ponto de entrada do componente

Um componente de serviço inicia esta função, durante a inicialização, para incluir um ponto de entrada no vetor do ponto de entrada para esse componente de serviço.

### Sintaxe

MQZEP ( *Hconfig* , *Function* , *EntryPoint* , *CompCode* , *Reason* )

### Parâmetros

#### Hconfig

Tipo: MQHCONFIG-entrada

Identificador de configuração Essa manipulação representa o componente que está sendo configurado para esse determinado serviço instalável Ele deve ser igual ao componente transmitido para a função de configuração do componente pelo gerenciador de fila na chamada de inicialização do componente.

#### Função

Tipo: MQLONG - entrada

Identificador de função. Valores válidos para isso são definidos para cada serviço instalável.

Se MQZEP for chamado mais de uma vez para a mesma função, a última chamada feita fornecerá o ponto de entrada usado.

#### EntryPoint

Tipo: PMQFUNC-entrada

Ponto de entrada da função Este é o endereço do ponto de entrada fornecido pelo componente para executar a função

O valor NULL é válido e indica que a função não é fornecida por esse componente. NULL é assumido para os pontos de entrada que não estão definidos usando MQZEP

#### CompCode

Tipo: MQLONG - saída

Código de conclusão. Deve ser um dos valores a seguir:

##### MQCC\_OK

Indica conclusão bem-sucedida.

##### MQCC\_FAILED

A chamada falhou.

#### Razão

Tipo: MQLONG - saída

Código de razão qualificando *CompCode*.

Se *CompCode* for MQCC\_OK:

##### MQRC\_NONE

(0, X'000') Nenhuma razão a relatar.

Se *CompCode* for MQCC\_FAILED:

##### MQRC\_FUNCTION\_ERROR

(2281, X'8E9') Identificador de função inválido.

##### MQRC\_HCONFIG\_ERROR

(2280, X'8E8') Identificador de configuração inválido.

Para obter mais informações sobre esses códigos de razão, consulte [conclusão da API e códigos de razão](#).

## Chamada C

```
MQZEP (Hconfig, Function, EntryPoint, &CompCode, &Reason);
```

Declare os parâmetros conforme a seguir:

```
MQHCONFIG  Hconfig;      /* Configuration handle */
MQLONG     Function;    /* Function identifier */
PMQFUNC    EntryPoint;  /* Function entry point */
MQLONG     CompCode;    /* Completion code */
MQLONG     Reason;     /* Reason code qualifying CompCode */
```

## MQZFP-Parâmetros livres

A estrutura MQZFP é usada na chamada MQZ\_FREE\_USER para o parâmetro *FreeParms* .. Este parâmetro especifica os dados relacionados ao recurso a ser liberado

[Tabela 1.](#) resume os campos na estrutura.

Tabela 841. Campos em MQZFP	
Campo	Descrição
<a href="#">StrucId</a>	Identificador de estruturação
<a href="#">Versão</a>	Versão
<a href="#">reservadas</a>	Campo reservado
<a href="#">CorrelationPtr</a>	Ponteiro de correlação

## Campos

### StrucId

Tipo: MQCHAR4 -entrada

Identificador de estruturação. O valor é o seguinte:

#### **MQZIC\_STRUC\_ID**

Identificador para a estrutura de contexto de identidade Para a linguagem de programação C, a constante MQZIC\_STRUC\_ID\_ARRAY também é definida; isso tem o mesmo valor que MQZIC\_STRUC\_ID, mas é uma matriz de caracteres em vez de uma sequência.

### Versão

Tipo: MQLONG - entrada

Número de versão da estrutura. O valor é o seguinte:

#### **MQZFP\_VERSION\_1**

Version-1 estrutura de parâmetros livres.

A constante a seguir especifica o número da versão atual:

#### **MQZFP\_CURRENT\_VERSION**

A versão atual da estrutura de parâmetros livres

### Reservado

Tipo: MQBYTE8 -entrada

Campo reservado.. O valor inicial é nulo.



## CorrelationPtr

Tipo: MQPTR-entrada

Indicador de correlação. Endereço de dados de correlações relacionados ao recurso a ser liberado

## Declaração C

```
typedef struct tagMQZFP MQZFP;  
struct tagMQZFP {  
    MQCHAR4    StrucId;           /* Structure identifier */  
    MQLONG     Version;          /* Structure version number */  
    MQBYTE8    Reserved;        /* Reserved field */  
    MQPTR      CorrelationPtr;   /* Address of correlation data */  
};
```

## MQZIC-Contexto de identidade

A estrutura MQZIC é usada na chamada MQZ\_AUTHENTICATE\_USER para o parâmetro *IdentityContext* ..

A estrutura MQZIC contém informações de contexto de identidade, que identifica o usuário do aplicativo que primeiro colocou a mensagem em uma fila:

- O gerenciador de filas preenche o campo *UserIdentifier* com um nome que identifica o usuário, a maneira como o gerenciador de filas pode fazer isso depende do ambiente no qual o aplicativo está em execução.
- O gerenciador de filas preenche o campo *AccountingToken* com um token ou número determinado a partir do aplicativo que colocou a mensagem.
- Os aplicativos podem usar o campo *AppIdentityData* para quaisquer informações extras que eles desejam incluir sobre o usuário (por exemplo, uma senha criptografada).

Os aplicativos adequadamente autorizados podem configurar o contexto de identidade usando a função MQZ\_AUTHENTICATE\_USER

Um Windows identificador de segurança do sistema (SID) é armazenado no campo *AccountingToken* quando uma mensagem é criada em IBM MQ for Windows. O SID pode ser usado para complementar o campo *UserIdentifier* e para estabelecer as credenciais de um usuário.

*Tabela 1.* resume os campos na estrutura.

Campo	Descrição
<a href="#">StrucId</a>	Identificador de estruturação
<a href="#">Versão</a>	Versão
<a href="#">UserIdentifier</a>	Identificador de usuários
<a href="#">AccountingToken</a>	Símbolo de contabilidade
<a href="#">AppIdentityData</a>	Dados de identidade de aplicativo

## Campos

### StrucId

Tipo: MQCHAR4 -entrada

Identificador de estruturação. O valor é o seguinte:

#### MQZIC\_STRUC\_ID

Identificador para a estrutura de contexto de identidade Para a linguagem de programação C, a constante MQZIC\_STRUC\_ID\_ARRAY também é definida; isso tem o mesmo valor que MQZIC\_STRUC\_ID, mas é uma matriz de caracteres em vez de uma sequência.

## Versão

Tipo: MQLONG - entrada

Número de versão da estrutura. O valor é o seguinte:

### **MQZIC\_VERSION\_1**

Version-1 estrutura de contexto de identidade.

A constante a seguir especifica o número da versão atual:

### **MQZIC\_CURRENT\_VERSION**

A versão atual da estrutura do contexto de identidade

## UserIdentifier

Tipo: MQCHAR12 - entrada

Identificador do usuário. Isso faz parte do contexto de identidade da mensagem *UserIdentifier* especifica o identificador do usuário do aplicativo que originou a mensagem. O gerenciador de filas trata essas informações como dados de caractere, mas não define o formato deles. Para obter mais informações sobre o campo *UserIdentifier*, consulte [“UserIdentifier \(MQCHAR12\) para MQMD” na página 473](#)

## AccountingToken

Tipo: MQBYTE32 - entrada

Token de Contabilidade. Isso faz parte do contexto de identidade da mensagem *AccountingToken* permite que um aplicativo faça com que o trabalho feito como resultado da mensagem seja adequadamente cobrado. O gerenciador de filas trata estas informações como uma sequência de bits e não verifica seu conteúdo. Para obter mais informações sobre o campo *AccountingToken*, consulte [“AccountingToken \(MQBYTE32\) para MQMD” na página 475](#)

## ApplIdentityData

Tipo: MQCHAR32 - entrada

Dados de aplicativos relacionados à identidade. Isso faz parte do contexto de identidade da mensagem *ApplIdentityDados* são informações definidas pelo conjunto de aplicativos que podem ser usadas para fornecer informações adicionais sobre a origem da mensagem. Por exemplo, ele poderia ser configurado por aplicativos em execução com autoridade de usuário adequada para indicar se os dados de identificação são confiáveis.. Para obter mais informações sobre o campo de dados *ApplIdentity*, consulte [“Dados de ApplIdentity\(MQCHAR32\) para MQMD” na página 477](#)

## Declaração C

```
typedef struct tagMQZED MQZED;
struct tagMQZED {
    MQCHAR4    StrucId;           /* Structure identifier */
    MQLONG     Version;         /* Structure version number */
    MQCHAR12   UserIdentifier;   /* User identifier */
    MQBYTE32   AccountingToken; /* Accounting token */
    MQCHAR32   ApplIdentityData; /* Application data relating to identity */
};
```



## Informações de referência da interface de serviços instaláveis no

### IBM i

Use estas informações para entender as informações de referência para os serviços instaláveis para o IBM i

Para cada função, há uma descrição, incluindo o identificador de função (para MQZEP)


Os *parâmetros* são mostrados na ordem em que devem ocorrer. Todos devem estar presentes.

Cada nome do parâmetro é seguido por seu tipo de dado entre parênteses Estes são os tipos de dados elementares descritos em [“Tipos de dados elementares” na página 1031](#)

A chamada de linguagem C também é fornecida, após a descrição dos parâmetros

### Conceitos relacionados

 [Serviços e componentes instaláveis para IBM i](#)

 [Serviços e componentes instaláveis para UNIX, Linux e Windows](#)

### Referências relacionadas

“Informações de referência da interface de serviços instaláveis” na página 1678  
Esta coleção de tópicos fornece informações de referência para os serviços instaláveis

## **MQZEP (Incluir ponto de entrada do componente) no IBM i**

Essa função é chamada por um componente de serviço, durante a inicialização, para incluir um ponto de entrada no vetor do ponto de entrada para esse componente de serviço.

### Sintaxe

```
MQZEP (Hconfig, Function, EntryPoint, CompCode, Reason)
```

### Parâmetros

A chamada MQZEP possui os seguintes parâmetros.

#### **Hconfig (MQHCONFIG)-entrada**

Identificador de configuração

Essa manipulação representa o componente que está sendo configurado para esse serviço instalável específico. Ele deve ser igual ao transmitido para a função de configuração do componente pelo gerenciador de fila na chamada de inicialização do componente.

#### **Função (MQLONG)-entrada**

Identificador de função.

Valores válidos para isso são definidos para cada serviço instalável. Se MQZEP for chamado mais de uma vez para a mesma função, a última chamada feita fornecerá o ponto de entrada usado.

#### **EntryPoint (PMQFUNC)-entrada**

Ponto de entrada da função

Este é o endereço do ponto de entrada fornecido pelo componente para executar a função. O valor NULL é válido e indica que a função não é fornecida por este componente. NULL é assumido para pontos de entrada que não são definidos usando MQZEP.

#### **CompCode (MQLONG)-saída**

Código de conclusão.

É um dos seguintes:

##### **MQCC\_OK**

Indica conclusão bem-sucedida.

##### **MQCC\_FAILED**

A chamada falhou.

#### **Motivo (MQLONG)-saída**

Código de razão qualificando *CompCode*.

Se *CompCode* for MQCC\_OK:

##### **MQRC\_NONE**

(0, X'000') Nenhuma razão a relatar.

Se *CompCode* for MQCC\_FAILED:

## **MQRC\_FUNCTION\_ERROR**

(2281, X'8E9') Identificador de função inválido.

## **MQRC\_HCONFIG\_ERROR**

(2280, X'8E8') Identificador de configuração inválido.

Para obter mais informações sobre esses códigos de razão, consulte [Mensagens e códigos de razão](#).

## **Chamada C**

```
MQZEP (Hconfig, Function, EntryPoint, &CompCode, &Reason);
```

Declare os parâmetros conforme a seguir:

```
MQHCONFIG  Hconfig;    /* Configuration handle */
MQQLONG    Function;   /* Function identifier */
PMQFUNC    EntryPoint; /* Function entry point */
MQQLONG    CompCode;   /* Completion code */
MQQLONG    Reason;     /* Reason code qualifying CompCode */
```

### **IBM i**

## **MQHCONFIG (Identificador de configuração) no IBM i**

O tipo de dados MQHCONFIG representa um identificador de configuração, ou seja, o componente que está sendo configurado para um serviço instalável específico. Uma alça de configuração deve ser alinhada em seu limite natural.

Os aplicativos devem testar variáveis desse tipo somente para igualdade..

## **Declaração C**

```
typedef void MQPOINTER MQHCONFIG;
```

### **IBM i**

## **PMQFUNC (Ponteiro para função) em IBM i**

Ponteiro para uma função.

## **Declaração C**

```
typedef void MQPOINTER PMQFUNC;
```

### **IBM i**

## **MQZ\_AUTHENTICATE\_USER (Autenticar usuário) no IBM i**

Essa função é fornecida por um componente de serviço de autorização MQZAS\_VERSION\_5 .. É chamado pelo gerenciador de filas para autenticar um usuário ou para configurar campos de contexto de identidade.

Ele é chamado quando um contexto de aplicativo de usuário do IBM MQ é estabelecido. Isso acontece durante as chamadas de conexão no ponto em que o contexto do usuário do aplicativo é inicializado e em cada ponto em que o contexto do usuário do aplicativo é alterado. Cada vez que uma chamada de conexão é feita, as informações de contexto do usuário do aplicativo são readquiridas no campo *IdentityContext*.

O identificador de função para esta função (para MQZEP) é MQZID\_AUTHENTICATE.

## **Sintaxe**

**MQZ\_AUTHENTICATE\_USER** (*QMgrName, SecurityParms, ApplicationContext, IdentityContext, CorrelationPtr, ComponentData, Continuation, CompCode, Reason*)

## Parâmetros

A chamada MQZ\_AUTHENTICATE\_USER possui os seguintes parâmetros.

### **QMgrName (MQCHAR48)-entrada de**

Nome do gerenciador de filas.

O nome do gerenciador de fila chamando o componente. Esse nome é preenchido com espaços em branco até o comprimento total do parâmetro; o nome não é terminado por um caractere nulo. O nome do gerenciador de filas é transmitido para o componente para obter informações; a interface de serviço de autorização não requer que o componente o use de nenhuma maneira definida

### **SecurityParms (MQCSP)-entrada**

Parâmetros de segurança..

Dados relacionados ao ID do usuário, senha e tipo de autenticação.

Durante uma chamada MQI MQCONN, esse parâmetro contém valores nulos ou padrão.

### **ApplicationContext (MQZAC)-entrada**

Contexto do aplicativo

Dados relacionados com a aplicação de chamada. Consulte [“MQZAC \(contexto de aplicativos\) no IBM i”](#) na página 1775 para obter detalhes. Durante cada chamada MQI MQCONN ou MQCONNX, as informações de contexto do usuário na estrutura MQZAC são readquiridas.

### **IdentityContext (MQZIC)-entrada/saída**

Contexto de identidade..

Na entrada para a função autenticar usuário, isso identifica o contexto de identidade atual. A função autenticar usuário pode mudar isso, nesse ponto o gerenciador de filas adota o novo contexto de identidade. Consulte [“MQZIC \(contexto de identidade\) no IBM i”](#) na página 1781 para obter mais detalhes sobre a estrutura do MQZIC

### **CorrelationPtr (MQPTR)-saída**

Indicador de correlação.

Especifica o endereço dos dados de correlação. Esse ponteiro é então transmitido para outras chamadas do OAM.

### **ComponentData (Comprimento de MQBYTE x ComponentData)-entrada/saída**

Dados do componente..

Esses dados são mantidos pelo gerenciador de filas em nome desse componente específico; quaisquer mudanças feitas nele por qualquer uma das funções fornecidas por esse componente são preservadas e apresentadas na próxima vez em que uma dessas funções de componentes for chamada.. O comprimento dessa área de dados é passado pelo gerenciador de filas no parâmetro **ComponentDataLength** da chamada MQZ\_INIT\_AUTHORITY

### **Continuação (MQLONG)-saída**

Sinalizador de continuação

Os valores a seguir podem ser especificados:

#### **MQZCI\_DEFAULT**

Continuação dependente de outros componentes..

#### **MQZCI\_STOP**

Não continue com o próximo componente

### **CompCode (MQLONG)-saída**

Código de conclusão.

É um dos seguintes:

**MQCC\_OK**

Indica conclusão bem-sucedida.

**MQCC\_FAILED**

A chamada falhou.

**Motivo (MQLONG)-saída**

Código de razão qualificando *CompCode*.

Se *CompCode* for MQCC\_OK:

**MQRC\_NONE**

(0, X'000') Nenhuma razão a relatar.

Se *CompCode* for MQCC\_FAILED:

**MQRC\_SERVICE\_ERROR**

(2289, X'8F1') Ocorreu um erro inesperado ao acessar o serviço.

Para obter mais informações sobre esses códigos de razão, consulte [Mensagens e códigos de razão](#).

## Chamada C

```
MQZ_AUTHENTICATE_USER (QMgrName, SecurityParms, ApplicationContext,  
                        IdentityContext, &CorrelationPtr, ComponentData,  
                        &Continuation, &CompCode, &Reason);
```

Os parâmetros transmitidos para o serviço são declarados da seguinte forma:

```
MQCHAR48  QMgrName;           /* Queue manager name */  
MQCSP     SecurityParms;      /* Security parameters */  
MQZAC     ApplicationContext; /* Application context */  
MQZIC     IdentityContext;    /* Identity context */  
MQPTR     CorrelationPtr;     /* Correlation pointer */  
MQBYTE    ComponentData[n];  /* Component data */  
MQLONG    Continuation;      /* Continuation indicator set by  
                             component */  
MQLONG    CompCode;          /* Completion code */  
MQLONG    Reason;           /* Reason code qualifying CompCode */
```

## IBM i MQZ\_CHECK\_AUTHORITY (Verificar autoridade) em IBM i .

Essa função é fornecida por um componente de serviço de autorização MQZAS\_VERSION\_1 e é chamada pelo gerenciador de filas para verificar se uma entidade tem autoridade para executar uma determinada ação, ou ações, em um objeto especificado...

O identificador de função para esta função (para MQZEP) é MQZID\_CHECK\_AUTHORITY.

## Sintaxe

**MQZ\_CHECK\_AUTHORITY (QMgrName, EntityName, EntityType, ObjectName, ObjectType, Authority, ComponentData, Continuation, CompCode, Reason)**

## Parâmetros

A chamada MQZ\_CHECK\_AUTHORITY possui os seguintes parâmetros.

**QMgrName (MQCHAR48)-entrada de**

Nome do gerenciador de filas.

O nome do gerenciador de fila chamando o componente. Esse nome é preenchido com espaços em branco até o comprimento total do parâmetro; o nome não é terminado por um caractere nulo. O

nome do gerenciador de filas é transmitido para o componente para obter informações; a interface de serviço de autorização não requer que o componente faça uso dele de nenhuma maneira definida

#### **EntityName (MQCHAR12)-entrada**

Nome da entidade..

O nome da entidade cuja autorização para o objeto deve ser verificada O comprimento máximo da sequência é de 12 caracteres; se ela for mais curta do que ela for preenchida à direita com espaços em branco. O nome não é terminado por um caractere nulo.

Não é essencial que esta entidade seja conhecida pelo serviço de segurança subjacente. Se não for conhecido, as autorizações do grupo especial **nobody** (ao qual todas as entidades são assumidas como pertencentes) serão usadas para a verificação. Um nome em branco é válido e pode ser usado dessa maneira.

#### **EntityType (MQLONG)-entrada**

Tipo de entidade..

O tipo de entidade especificado por *EntityName* É um dos seguintes:

##### **MQZAET\_PRINCIPAL**

Principal.

##### **MQZAET\_GROUP**

#### **ObjectName (MQCHAR48)-entrada de**

Nome do objeto.

O nome do objeto para o qual o acesso é necessário O comprimento máximo da sequência é de 48 caracteres; se for mais curto do que isso, será preenchido à direita com espaços em branco. O nome não é terminado por um caractere nulo.

Se *ObjectType* for MQOT\_Q\_MGR, esse nome será o mesmo que *QMgrName*

#### **ObjectType (MQLONG)-entrada**

Tipo de objeto.

O tipo de entidade especificado por *ObjectName* É um dos seguintes:

##### **MQOT\_AUTH\_INFO**

Informações de autenticação..

##### **MQOT\_CHANNEL**

Canal.

##### **MQOT\_CLNTCONN\_CHANEXO L**

Canal de conexão do cliente.

##### **MQOT\_LISTENER**

Ouvinte.

##### **MQOT\_NAMELIST**

Lista de nomes

##### **MQOT\_PROCESS**

process definition.

##### **MQOT\_Q**

Fila.

##### **MQOT\_Q\_MGR**

Gerenciador de Filas

##### **MQOT\_SERVICE**

Serviço.

#### **Autoridade (MQLONG)-entrada**

A autoridade a ser verificada

Se uma autorização estiver sendo verificada, este campo será igual à operação de autorização apropriada (constante MQZAO\_\*) Se mais de uma autorização estiver sendo verificada, ela será OR bit a bit das constantes MQZAO\_\* correspondentes.

As autorizações a seguir se aplicam ao uso das chamadas MQI:

#### **MQZAO\_CONNECT**

Capacidade de usar a chamada MQCONN

#### **MQZAO\_BROWSE**

Capacidade de usar a chamada MQGET com uma opção de procura

Isso permite que a opção MQGMO\_BROWSE\_FIRST, MQGMO\_BROWSE\_MSG\_UNDER\_CURSOR ou MQGMO\_BROWSE\_NEXT seja especificada na chamada MQGET..

#### **MQZAO\_INPUT**

Capacidade de usar a chamada MQGET com uma opção de entrada

Isso permite que a opção MQOO\_INPUT\_SHARED, MQOO\_INPUT\_EXCLUSIVE ou MQOO\_INPUT\_AS\_Q\_DEF seja especificada na chamada MQOPEN.

#### **MQZAO\_OUTPUT**

Capacidade de usar a chamada MQPUT

Isso permite que a opção MQOO\_OUTPUT seja especificada na chamada MQOPEN..

#### **MQZAO\_INQUIRE**

Capacidade de usar a chamada MQINQ..

Isso permite que a opção MQOO\_INQUIRE seja especificada na chamada MQOPEN..

#### **MQZAO\_SET**

Capacidade de usar a chamada MQSET

Isso permite que a opção MQOO\_SET seja especificada na chamada MQOPEN..

#### **MQZAO\_PASS\_IDENTITY\_CONTEXT**

Capacidade de passar o contexto de identidade

Isso permite que a opção MQOO\_PASS\_IDENTITY\_CONTEXT seja especificada na chamada MQOPEN e que a opção MQPMO\_PASS\_IDENTITY\_CONTEXT seja especificada nas chamadas MQPUT e MQPUT1 .

#### **MQZAO\_PASS\_ALL\_CONTEXT**

Capacidade de passar todo o contexto

Isso permite que a opção MQOO\_PASS\_ALL\_CONTEXT seja especificada na chamada MQOPEN e a opção MQPMO\_PASS\_ALL\_CONTEXT seja especificada nas chamadas MQPUT e MQPUT1 .

#### **MQZAO\_SET\_IDENTITY\_CONTEXT**

Capacidade de configurar o contexto de identidade

Isso permite que a opção MQOO\_SET\_IDENTITY\_CONTEXT seja especificada na chamada MQOPEN e a opção MQPMO\_SET\_IDENTITY\_CONTEXT seja especificada nas chamadas MQPUT e MQPUT1 .

#### **MQZAO\_SET\_ALL\_CONTEXT**

Capacidade de configurar todo o contexto

Isso permite que a opção MQOO\_SET\_ALL\_CONTEXT seja especificada na chamada MQOPEN e a opção MQPMO\_SET\_ALL\_CONTEXT seja especificada nas chamadas MQPUT e MQPUT1 .

#### **MQZAO\_ALTERNATE\_USER\_AUTHORITY**

Capacidade de usar a autoridade de usuário alternativa

Isso permite que a opção MQOO\_ALTERNATE\_USER\_AUTHORITY seja especificada na chamada MQOPEN, e a opção MQPMO\_ALTERNATE\_USER\_AUTHORITY seja especificada na chamada MQPUT1 .



**MQZAO\_ALL\_MQI**

Todas as autorizações de MQI

Isso ativa todas as autorizações descritas anteriormente..

As autorizações a seguir se aplicam à administração de um gerenciador de filas:

**MQZAO\_CREATE**

Capacidade de criar objetos de um tipo especificado

**MQZAO\_DELETE**

Capacidade de excluir um objeto especificado..

**MQZAO\_DISPLAY**

Capacidade de exibir os atributos de um objeto especificado..

**MQZAO\_CHANGE**

A capacidade de alterar os atributos de um objeto especificado.

**MQZAO\_CLEAR**

Capacidade de excluir todas as mensagens de uma fila especificada.

**MQZAO\_AUTORIZAR**

Capacidade de autorizar outros usuários para um objeto especificado..

**MQZAO\_CONTROL**

Capacidade de iniciar, parar ou efetuar ping de um objeto de canal não cliente...

**MQZAO\_CONTROL\_EXTENDED**

Capacidade de reconfigurar um número de sequência ou resolver uma mensagem pendente em um objeto de canal não cliente.

**MQZAO\_ALL\_ADMIN**

Todas as autorizações de administração, além de MQZAO\_CREATE

As autorizações a seguir se aplicam ao uso do MQI e à administração de um gerenciador de filas:

**MQZAO\_ALL**

Todas as autorizações, além de MQZAO\_CREATE

**MQZAO\_NONE**

Sem autorizações.

**ComponentData (Comprimento de MQBYTE x ComponentData)-entrada/saída**

Dados do componente..

Esses dados são mantidos pelo gerenciador de filas em nome desse componente específico; quaisquer mudanças feitas nele por qualquer uma das funções fornecidas por esse componente são preservadas e apresentadas na próxima vez em que uma das funções desse componente for chamada de.

O comprimento dessa área de dados é passado pelo gerenciador de filas no parâmetro

**ComponentDataLength** da chamada MQZ\_INIT\_AUTHORITY

**Continuação (MQLONG)-saída**

Indicador de continuação configurado por componente..

Os valores a seguir podem ser especificados:

**MQZCI\_DEFAULT**

Continuação dependente do gerenciador de filas..

Para MQZ\_CHECK\_AUTHORITY, isso tem o mesmo efeito que MQZCI\_STOP

**MQZCI\_CONTINUE**

Continue com o próximo componente

**MQZCI\_STOP**

Não continue com o próximo componente

### CompCode (MQLONG)-saída

Código de conclusão.

É um dos seguintes:

#### **MQCC\_OK**

Indica conclusão bem-sucedida.

#### **MQCC\_FAILED**

A chamada falhou.

### Motivo (MQLONG)-saída

Código de razão qualificando *CompCode*.

Se *CompCode* for MQCC\_OK:

#### **MQRC\_NONE**

(0, X'000') Nenhuma razão a relatar.

Se *CompCode* for MQCC\_FAILED:

#### **MQRC\_NOT\_AUTHORIZED**

(2035, X'7F3') Não autorizado para acesso.

#### **MQRC\_SERVICE\_ERROR**

(2289, X'8F1') Ocorreu um erro inesperado ao acessar o serviço.

#### **MQRC\_SERVICE\_NOT\_AVAILABLE**

(2285, X'8ED') Serviço subjacente não disponível.

Para obter mais informações sobre esses códigos de razão, consulte [Mensagens e códigos de razão](#).

## Chamada C

```
MQZ_CHECK_AUTHORITY (QMgrName, EntityName, EntityType, ObjectName,  
                    ObjectType, Authority, ComponentData,  
                    &Continuation, &CompCode, &Reason);
```

Os parâmetros transmitidos para o serviço são declarados da seguinte forma:

```
MQCHAR48  QMgrName;           /* Queue manager name */  
MQCHAR12  EntityName;        /* Entity name */  
MQLONG    EntityType;        /* Entity type */  
MQCHAR48  ObjectName;       /* Object name */  
MQLONG    ObjectType;       /* Object type */  
MQLONG    Authority;        /* Authority to be checked */  
MQBYTE    ComponentData[n]; /* Component data */  
MQLONG    Continuation;     /* Continuation indicator set by  
                             component */  
MQLONG    CompCode;        /* Completion code */  
MQLONG    Reason;         /* Reason code qualifying CompCode */
```

### **MQZ\_CHECK\_PRIVILEGED-Verifique se o usuário é privilegiado**

Essa função é fornecida por um componente de serviço de autorização MQZAS\_VERSION\_6 e é chamada pelo gerenciador de filas para determinar se um usuário especificado é um usuário privilegiado.

O identificador de função para essa função (para MQZEP) é MQZID\_CHECK\_PRIVILEGED.

### Sintaxe

```
MQZ_CHECK_PRIVILEGED( QMgrName , EntityData , EntityType , ComponentData ,  
Continuation , CompCode , Reason )
```

## Parâmetros

### QMgrName

Tipo: MQCHAR48 - entrada

Nome do gerenciador de filas. O nome do gerenciador de fila chamando o componente. Esse nome é preenchido com espaços em branco até o comprimento total do parâmetro; o nome não é terminado por um caractere nulo.

O nome do gerenciador de filas é transmitido para o componente para obter informações; a interface de serviço de autorização não requer que o componente o use de nenhuma maneira definida

### EntityData

Tipo: MQZED-entrada

Dados da entidade.. Dados relacionados à entidade que será verificada. Para obter mais informações, consulte [“MQZED-Descritor de entidade” na página 1737](#).

### EntityType

Tipo: MQLONG - entrada

Tipo de entidade.. O tipo de entidade especificado por EntityData Deve ser um dos valores a seguir:

#### **MQZAET\_PRINCIPAL**

Principal.

#### **MQZAET\_GROUP**

### ComponentData

Tipo: MQBYTEComponentDataLength -entrada/saída.

Dados do componente.. Esses dados são mantidos pelo gerenciador de filas em nome desse componente específico; quaisquer mudanças feitas nele por qualquer uma das funções fornecidas por esse componente são preservados e apresentados na próxima vez em que uma dessas funções do componente for chamada

O comprimento dessa área de dados é passado pelo gerenciador de filas no parâmetro **ComponentDataLength** da chamada MQZ\_INIT\_AUTHORITY

### Continuação

Tipo: MQLONG - saída

Indicador de continuação configurado por componente.. Os valores a seguir podem ser especificados:

#### **MQZCI\_DEFAULT**

Continuação dependente do gerenciador de filas..

Para MQZ\_CHECK\_AUTHORITY, isso tem o mesmo efeito que MQZCI\_STOP

#### **MQZCI\_CONTINUE**

Continue com o próximo componente

#### **MQZCI\_STOP**

Não continue com o próximo componente

Se a chamada para um componente falhar (ou seja *CompCode* retorna MQCC\_FAILED) e o parâmetro *Continuation* for MQZCI\_DEFAULT ou MQZCI\_CONTINUE, o gerenciador de filas continuará a chamar outros componentes se houver algum.

Se a chamada for bem-sucedida (ou seja, *CompCode* retorna MQCC\_OK), nenhum outro componente será chamado, independentemente da configuração de *Continuação* .

Se a chamada falhar e o parâmetro *Continuation* for MQZCI\_STOP, nenhum outro componente será chamado e o erro será retornado ao gerenciador de fila. Os componentes não têm conhecimento de chamadas anteriores, portanto, o parâmetro *Continuation* é sempre configurado como MQZCI\_DEFAULT antes da chamada.

## CompCode

Tipo: MQLONG - saída

Código de conclusão. Deve ser um dos valores a seguir:

### **MQCC\_OK**

Indica conclusão bem-sucedida.

### **MQCC\_FAILED**

A chamada falhou.

## Razão

Tipo: MQLONG - saída

Código de razão qualificando *CompCode*.

Se *CompCode* for MQCC\_OK:

### **MQRC\_NONE**

(0, X'000') Nenhuma razão a relatar.

Se *CompCode* for MQCC\_FAILED:

### **MQRC\_NOT\_PRIVILEGED**

(2584, X'A18') Este usuário não é um ID do usuário privilegiado

### **MQRC\_UNKNOWN\_ENTITY**

(2292, X'8F4') Entidade desconhecida para serviço.

### **MQRC\_SERVICE\_ERROR**

(2289, X'8F1') Ocorreu um erro inesperado ao acessar o serviço.

### **MQRC\_SERVICE\_NOT\_AVAILABLE**

(2285, X'8ED') Serviço subjacente não disponível.

Para obter mais informações sobre esses códigos de razão, consulte [conclusão da API e códigos de razão](#).

## Chamada C

```
MQZ_CHECK_PRIVILEGED (QMgrName, &EntityData, EntityType,  
                     ComponentData, &Continuation,  
                     &CompCode, &Reason);
```

Os parâmetros transmitidos para o serviço são declarados da seguinte forma:

```
MQCHAR48  QMgrName;           /* Queue manager name */  
MQZED     EntityData;        /* Entity name */  
MQLONG    EntityType;       /* Entity type */  
MQBYTE    ComponentData[n]; /* Component data */  
MQLONG    Continuation;     /* Continuation indicator set by  
                           component */  
MQLONG    CompCode;         /* Completion code */  
MQLONG    Reason;          /* Reason code qualifying CompCode */
```

## **MQZ\_COPY\_ALL\_AUTHORITY (Copiar todas as autoridades) em IBM i**

Essa função é fornecida por um componente de serviço de autorização. Ele é chamado pelo gerenciador de filas para copiar todas as autorizações que estão atualmente em vigor para um objeto de referência para outro objeto.

O identificador de função para esta função (para MQZEP) é MQZID\_COPY\_ALL\_AUTHORITY.

## Sintaxe

**MQZ\_COPY\_ALL\_AUTHORITY** (*QMgrName, RefObjectName, ObjectName, ObjectType, ComponentData, Continuation, CompCode, Reason*)

## Parâmetros

A chamada MQZ\_COPY\_ALL\_AUTHORITY tem os seguintes parâmetros.

### **QMgrName (MQCHAR48)-entrada de**

Nome do gerenciador de filas.

O nome do gerenciador de fila chamando o componente. Esse nome é preenchido com espaços em branco até o comprimento total do parâmetro; o nome não é terminado por um caractere nulo.

O nome do gerenciador de filas é transmitido para o componente para obter informações; a interface de serviço de autorização não requer que o componente faça uso dele de nenhuma maneira definida

### **RefObjectName (MQCHAR48)-entrada**

Nome do objeto de referencia

O nome do objeto de referência, cujas autorizações devem ser copiadas. O comprimento máximo da sequência é de 48 caracteres; se for mais curto do que isso, será preenchido à direita com espaços em branco. O nome não é terminado por um caractere nulo.

### **ObjectName (MQCHAR48)-entrada de**

Nome do objeto.

O nome do objeto para o qual os acessos devem ser configurados O comprimento máximo da sequência é de 48 caracteres; se for mais curto do que isso, será preenchido à direita com espaços em branco. O nome não é terminado por um caractere nulo.

### **ObjectType (MQLONG)-entrada**

Tipo de objeto.

O tipo de objeto especificado por *RefObjectName* e *ObjectName*.. É um dos seguintes:

#### **MQOT\_AUTH\_INFO**

Informações de autenticação..

#### **MQOT\_CHANNEL**

Canal.

#### **MQOT\_CLNTCONN\_CHANEXOL**

Canal de conexão do cliente.

#### **MQOT\_LISTENER**

Ouvinte.

#### **MQOT\_NAMELIST**

Lista de nomes

#### **MQOT\_PROCESS**

process definition.

#### **MQOT\_Q**

Fila.

#### **MQOT\_Q\_MGR**

Gerenciador de Filas

#### **MQOT\_SERVICE**

Serviço.

### **ComponentData (Comprimento de MQBYTE x ComponentData)-entrada/saída**

Dados do componente..

Esses dados são mantidos pelo gerenciador de filas em nome desse componente específico; quaisquer mudanças feitas nele por qualquer uma das funções fornecidas por esse componente são preservadas e apresentadas na próxima vez em que uma das funções desse componente for chamada de.

O comprimento dessa área de dados é passado pelo gerenciador de filas no parâmetro **ComponentDataLength** da chamada MQZ\_INIT\_AUTHORITY

### Continuação (MQLONG)-saída

Indicador de continuação configurado por componente..

Os valores a seguir podem ser especificados:

#### **MQZCI\_DEFAULT**

Continuação dependente do gerenciador de filas..

Para MQZ\_COPY\_ALL\_AUTHORITY, isso tem o mesmo efeito que MQZCI\_STOP

#### **MQZCI\_CONTINUE**

Continue com o próximo componente

#### **MQZCI\_STOP**

Não continue com o próximo componente

### CompCode (MQLONG)-saída

Código de conclusão.

É um dos seguintes:

#### **MQCC\_OK**

Indica conclusão bem-sucedida.

#### **MQCC\_FAILED**

A chamada falhou.

### Motivo (MQLONG)-saída

Código de razão qualificando *CompCode*.

Se *CompCode* for MQCC\_OK:

#### **MQRC\_NONE**

(0, X'000') Nenhuma razão a relatar.

Se *CompCode* for MQCC\_FAILED:

#### **MQRC\_SERVICE\_ERROR**

(2289, X'8F1') Ocorreu um erro inesperado ao acessar o serviço.

#### **MQRC\_SERVICE\_NOT\_AVAILABLE**

(2285, X'8ED') Serviço subjacente não disponível.

#### **MQRC\_UNKNOWN\_REF\_OBJECT**

(2294, X'8F6') Objeto de referência desconhecido.

Para obter mais informações sobre esses códigos de razão, consulte [Mensagens e códigos de razão](#).

## Chamada C

```
MQZ_COPY_ALL_AUTHORITY (QMgrName, RefObjectName, ObjectName, ObjectType,  
                        ComponentData, &Continuation, &CompCode,  
                        &Reason);
```

Os parâmetros transmitidos para o serviço são declarados da seguinte forma:

```
MQCHAR48 QMgrName;          /* Queue manager name */  
MQCHAR48 RefObjectName;     /* Reference object name */  
MQCHAR48 ObjectName;       /* Object name */  
MQLONG   ObjectType;       /* Object type */  
MQBYTE   ComponentData[n]; /* Component data */  
MQLONG   Continuation;     /* Continuation indicator set by  
                           component */  
MQLONG   CompCode;        /* Completion code */  
MQLONG   Reason;         /* Reason code qualifying CompCode */
```

Esta função é fornecida por um componente de serviço de autorização e é chamada pelo gerenciador de filas para excluir todas as autorizações associadas ao objeto especificado...

O identificador de função para esta função (para MQZEP) é MQZID\_DELETE\_AUTHORITY.

## Sintaxe

**MQZ\_DELETE\_AUTHORITY** (*QMgrName, ObjectName, ObjectType, ComponentData, Continuation, CompCode, Reason*)

## Parâmetros

A chamada MQZ\_DELETE\_AUTHORITY possui os seguintes parâmetros.

### **QMgrName (MQCHAR48)-entrada de**

Nome do gerenciador de filas.

O nome do gerenciador de fila chamando o componente. Esse nome é preenchido com espaços em branco até o comprimento total do parâmetro; o nome não é terminado por um caractere nulo.

O nome do gerenciador de filas é transmitido para o componente para obter informações; a interface de serviço de autorização não requer que o componente faça uso dele de nenhuma maneira definida

### **ObjectName (MQCHAR48)-entrada de**

Nome do objeto.

O nome do objeto para o qual os acessos devem ser excluídos O comprimento máximo da sequência é de 48 caracteres; se for mais curto do que isso, será preenchido à direita com espaços em branco. O nome não é terminado por um caractere nulo.

Se *ObjectType* for MQOT\_Q\_MGR, esse nome será o mesmo que *QMgrName*

### **ObjectType (MQLONG)-entrada**

Tipo de objeto.

O tipo de entidade especificado por *ObjectName* É um dos seguintes:

#### **MQOT\_AUTH\_INFO**

Informações de autenticação..

#### **MQOT\_CHANNEL**

Canal.

#### **MQOT\_CLNTCONN\_CHANEXO L**

Canal de conexão do cliente.

#### **MQOT\_LISTENER**

Ouvinte.

#### **MQOT\_NAMELIST**

Lista de nomes

#### **MQOT\_PROCESS**

process definition.

#### **MQOT\_Q**

Fila.

#### **MQOT\_Q\_MGR**

Gerenciador de Filas

#### **MQOT\_SERVICE**

Serviço.

### **ComponentData (Comprimento de MQBYTE x ComponentData)-entrada/saída**

Dados do componente..

Esses dados são mantidos pelo gerenciador de filas em nome desse componente específico; quaisquer mudanças feitas nele por qualquer uma das funções fornecidas por esse componente são preservadas e apresentadas na próxima vez em que uma das funções desse componente for chamada de.

O comprimento dessa área de dados é passado pelo gerenciador de filas no parâmetro **ComponentDataLength** da chamada MQZ\_INIT\_AUTHORITY

### Continuação (MQLONG)-saída

Indicador de continuação configurado por componente..

Os valores a seguir podem ser especificados:

#### **MQZCI\_DEFAULT**

Continuação dependente do gerenciador de filas..

Para MQZ\_DELETE\_AUTHORITY, isso tem o mesmo efeito que MQZCI\_STOP

#### **MQZCI\_CONTINUE**

Continue com o próximo componente

#### **MQZCI\_STOP**

Não continue com o próximo componente

### CompCode (MQLONG)-saída

Código de conclusão.

É um dos seguintes:

#### **MQCC\_OK**

Indica conclusão bem-sucedida.

#### **MQCC\_FAILED**

A chamada falhou.

### Motivo (MQLONG)-saída

Código de razão qualificando *CompCode*.

Se *CompCode* for MQCC\_OK:

#### **MQRC\_NONE**

(0, X'000') Nenhuma razão a relatar.

Se *CompCode* for MQCC\_FAILED:

#### **MQRC\_SERVICE\_ERROR**

(2289, X'8F1') Ocorreu um erro inesperado ao acessar o serviço.

#### **MQRC\_SERVICE\_NOT\_AVAILABLE**

(2285, X'8ED') Serviço subjacente não disponível.

Para obter mais informações sobre esses códigos de razão, consulte [Mensagens e códigos de razão](#).

## Chamada C

```
MQZ_DELETE_AUTHORITY (QMgrName, ObjectName, ObjectType, ComponentData,  
                      &Continuation, &CompCode, &Reason);
```

Os parâmetros transmitidos para o serviço são declarados da seguinte forma:

```
MQCHAR48 QMgrName;          /* Queue manager name */  
MQCHAR48 ObjectName;       /* Object name */  
MQLONG   ObjectType;       /* Object type */  
MQBYTE   ComponentData[n]; /* Component data */  
MQLONG   Continuation;     /* Continuation indicator set by  
                           component */  
MQLONG   CompCode;        /* Completion code */  
MQLONG   Reason;         /* Reason code qualifying CompCode */
```



## MQZ\_ENUMERATE\_AUTHORITY\_DATA (Enumerar dados de autoridade) em IBM i

Essa função é fornecida por um componente de serviço de autorização MQZAS\_VERSION\_4 e é chamada repetidamente pelo gerenciador de filas para recuperar todos os dados de autoridade que correspondem aos critérios de seleção especificados na primeira invocação..

O identificador de função para essa função (para MQZEP) é MQZID\_ENUMERATE\_AUTHORITY\_DATA.

### Sintaxe

**MQZ\_ENUMERATE\_AUTHORITY\_DATA** (*QMgrName*, *StartEnumeration*, *Filter*, *AuthorityBufferLength*, *AuthorityBuffer*, *AuthorityDataLength*, *ComponentData*, *Continuation*, *CompCode*, *Reason*)

### Parâmetros

A chamada MQZ\_ENUMERATE\_AUTHORITY\_DATA tem os seguintes parâmetros.

#### QMgrName (MQCHAR48)-entrada de

Nome do gerenciador de filas.

O nome do gerenciador de fila chamando o componente. Esse nome é preenchido com espaços em branco até o comprimento total do parâmetro; o nome não é terminado por um caractere nulo.

O nome do gerenciador de filas é transmitido para o componente para obter informações; a interface de serviço de autorização não requer que o componente faça uso dele de nenhuma maneira definida

#### StartEnumeration (MQLONG)-entrada

Sinalizador que indica se a chamada deve iniciar a enumeração

Isso indica se a chamada deve iniciar a enumeração de dados de autoridade ou continuar a enumeração de dados de autoridade iniciados por uma chamada anterior para MQZ\_ENUMERATE\_AUTHORITY\_DATA O valor é um dos seguintes:

##### MQZSE\_START

Inicie a enumeração

A chamada é chamada com esse valor para iniciar a enumeradas de dados de autoridade. O parâmetro **Filter** especifica os critérios de seleção a serem usados para selecionar dados de autoridade retornados por esta e chamadas sucessivas.

##### MQZSE\_CONTINUE

Continuar a enumeração

A chamada é chamado com este valor para continuar a enumeração de dados de autoridade. O parâmetro **Filter** é ignorado neste caso e pode ser especificado como ponteiro nulo (os critérios de seleção são determinados pelo parâmetro **Filter** especificado pela chamada que tinha *StartEnumeration* configurado como MQZSE\_START).

#### Filtro (MQZAD)-entrada

Filtro.

Se *StartEnumeration* for MQZSE\_START, *Filter* especifica os critérios de seleção a serem usados para selecionar os dados de autoridade a serem retornados. Se *Filter* for o ponteiro nulo, nenhum critério de seleção será usado, ou seja, todos os dados de autoridade serão retornados.. Consulte [“MQZAD \(Dados de autoridade\) no IBM i” na página 1777](#) para obter detalhes sobre os critérios de seleção que podem ser usados

Se *StartEnumeration* for MQZSE\_CONTINUE, *Filter* será ignorado e poderá ser especificado como o ponteiro nulo.

#### Comprimento de AuthorityBuffer(MQLONG)-entrada

O comprimento de *AuthorityBuffer*

Esse é o comprimento em bytes do parâmetro **AuthorityBuffer** . O buffer de autoridade deve ser grande o suficiente para acomodar os dados a serem retornadas

#### **AuthorityBuffer (MQZAD)-saída**

Dados de autoridade

Este é o buffer no qual os dados de autoridade são retornadas O buffer deve ser grande o suficiente para acomodar uma estrutura MQZAD, uma estrutura MQZED, mais o nome de entidade mais longo e o nome de domínio mais longo definido

**Nota:** Esse parâmetro é definido como um MQZAD, pois o MQZAD sempre ocorre no início do buffer.. No entanto, se o buffer for realmente declarado como um MQZAD, o buffer será muito pequeno-ele precisa ser maior do que um MQZAD para que possa acomodar o MQZAD, MQZED, além de nomes de entidade e domínio.

#### **AuthorityDataComprimento (MQLONG)-saída**

O comprimento dos dados retornados em *AuthorityBuffer*

Este é o comprimento dos dados retornados em *AuthorityBuffer* Se o buffer de autoridade for muito pequeno, o *AuthorityDataLength* será configurado para o comprimento do buffer necessário e a chamada retornará o código de conclusão MQCC\_FAILED e o código de razão MQRC\_BUFFER\_LENGTH\_ERROR.

#### **ComponentData (Comprimento de MQBYTE x ComponentData)-entrada/saída**

Dados do componente..

Esses dados são mantidos pelo gerenciador de filas em nome desse componente específico; quaisquer mudanças feitas nele por qualquer uma das funções fornecidas por esse componente são preservadas e apresentadas na próxima vez em que uma das funções desse componente for chamada de.

O comprimento dessa área de dados é passado pelo gerenciador de filas no parâmetro

**ComponentDataLength** da chamada MQZ\_INIT\_AUTHORITY

#### **Continuação (MQLONG)-saída**

Indicador de continuação configurado por componente..

Os valores a seguir podem ser especificados:

##### **MQZCI\_DEFAULT**

Continuação dependente do gerenciador de filas..

Para MQZ\_ENUMERATE\_AUTHORITY\_DATA, isso tem o mesmo efeito que MQZCI\_CONTINUE

##### **MQZCI\_CONTINUE**

Continue com o próximo componente

##### **MQZCI\_STOP**

Não continue com o próximo componente

#### **CompCode (MQLONG)-saída**

Código de conclusão.

É um dos seguintes:

##### **MQCC\_OK**

Indica conclusão bem-sucedida.

##### **MQCC\_FAILED**

A chamada falhou.

#### **Motivo (MQLONG)-saída**

Código de razão qualificando *CompCode*.

Se *CompCode* for MQCC\_OK:

##### **MQRC\_NONE**

(0, X'000') Nenhuma razão a relatar.

Se *CompCode* for MQCC\_FAILED:

#### **MQRC\_BUFFER\_LENGTH\_ERROR**

(2005, X'7D5') Parâmetro de comprimento de buffer inválido.

#### **MQRC\_NO\_DATA\_AVAILABLE**

(2379, X'94B') Não há dados disponíveis.

#### **MQRC\_SERVICE\_ERROR**

(2289, X'8F1') Ocorreu um erro inesperado ao acessar o serviço.

Para obter mais informações sobre esses códigos de razão, consulte [Mensagens e códigos de razão](#).

## Chamada C

```
MQZ_ENUMERATE_AUTHORITY_DATA (QMgrName, StartEnumeration, &Filter,  
    AuthorityBufferLength,  
    &AuthorityBuffer,  
    &AuthorityDataLength, ComponentData,  
    &Continuation, &CompCode,  
    &Reason);
```

Os parâmetros transmitidos para o serviço são declarados da seguinte forma:

```
MQCHAR48  QMgrName;           /* Queue manager name */  
MQLONG    StartEnumeration;  /* Flag indicating whether call should  
                               start enumeration */  
  
MQZAD     Filter;           /* Filter */  
MQLONG    AuthorityBufferLength; /* Length of AuthorityBuffer */  
MQZAD     AuthorityBuffer;  /* Authority data */  
MQLONG    AuthorityDataLength; /* Length of data returned in  
                               AuthorityBuffer */  
  
MQBYTE    ComponentData[n]; /* Component data */  
MQLONG    Continuation;     /* Continuation indicator set by  
                               component */  
  
MQLONG    CompCode;         /* Completion code */  
MQLONG    Reason;          /* Reason code qualifying CompCode */
```

## MQZ\_FREE\_USER-Usuário Livre

Essa função é fornecida por um componente de serviço de autorização MQZAS\_VERSION\_5 e é chamado pelo gerenciador de filas para liberar o recurso alocado associado. Ele é chamado quando um aplicativo conclui a execução em todos os contextos do usuário, por exemplo, durante uma chamada MQI MQDISC

O identificador de função para essa função (para MQZEP) é MQZID\_FREE\_USER

## MQZ\_GET\_AUTHORITY (obter autoridade) em IBM i

Essa função é fornecida por um componente de serviço de autorização MQZAS\_VERSION\_1 e é chamada pelo gerenciador de filas para recuperar a autoridade que uma entidade tem para acessar o objeto especificado.

O identificador de função para essa função (para MQZEP) é MQZID\_GET\_AUTHORITY

## Sintaxe

**MQZ\_GET\_AUTHORITY** (*QMgrName, EntityName, EntityType, ObjectName, ObjectType, Authority, ComponentData, Continuation, CompCode, Reason*)

## Parâmetros

A chamada MQZ\_GET\_AUTHORITY possui os seguintes parâmetros.

### **QMgrName (MQCHAR48)-entrada de**

Nome do gerenciador de filas.

O nome do gerenciador de fila chamando o componente. Esse nome é preenchido com espaços em branco até o comprimento total do parâmetro; o nome não é terminado por um caractere nulo.

O nome do gerenciador de filas é transmitido para o componente para obter informações; a interface de serviço de autorização não requer que o componente faça uso dele de nenhuma maneira definida

#### **EntityName (MQCHAR12)-entrada**

Nome da entidade..

O nome da entidade cujo acesso ao objeto deve ser recuperado. O comprimento máximo da sequência é de 12 caracteres; se ela for mais curta do que ela for preenchida à direita com espaços em branco. O nome não é terminado por um caractere nulo.

#### **EntityType (MQLONG)-entrada**

Tipo de entidade..

O tipo de entidade especificado por *EntityName* O valor a seguir pode ser especificado:

##### **MQZAET\_PRINCIPAL**

Principal.

##### **MQZAET\_GROUP**

#### **ObjectName (MQCHAR48)-entrada de**

Nome do objeto.

O nome do objeto para o qual a autoridade da entidade deve ser recuperada O comprimento máximo da sequência é de 48 caracteres; se for mais curto do que isso, será preenchido à direita com espaços em branco. O nome não é terminado por um caractere nulo.

Se *ObjectType* for MQOT\_Q\_MGR, esse nome será o mesmo que *QMgrName*

#### **ObjectType (MQLONG)-entrada**

Tipo de objeto.

O tipo de entidade especificado por *ObjectName* É um dos seguintes:

##### **MQOT\_AUTH\_INFO**

Informações de autenticação..

##### **MQOT\_CHANNEL**

Canal.

##### **MQOT\_CLNTCONN\_CHANEXO L**

Canal de conexão do cliente.

##### **MQOT\_LISTENER**

Ouvinte.

##### **MQOT\_NAMELIST**

Lista de nomes

##### **MQOT\_PROCESS**

process definition.

##### **MQOT\_Q**

Fila.

##### **MQOT\_Q\_MGR**

Gerenciador de Filas

##### **MQOT\_SERVICE**

Serviço.

#### **Autoridade (MQLONG)-saída**

Autoridade da entidade..

Se a entidade tiver uma autoridade, esse campo será igual à operação de autorização apropriada (constante MQZAO\_\*).. Se tiver mais de uma autoridade, esse campo será o OR bit a bit das constantes MQZAO\_\* correspondentes.

### **ComponentData (Comprimento de MQBYTE x ComponentData)-entrada/saída**

Dados do componente..

Esses dados são mantidos pelo gerenciador de filas em nome desse componente específico; quaisquer mudanças feitas nele por qualquer uma das funções fornecidas por esse componente são preservadas e apresentadas na próxima vez em que uma das funções desse componente for chamada de.

O comprimento dessa área de dados é passado pelo gerenciador de filas no parâmetro **ComponentDataLength** da chamada MQZ\_INIT\_AUTHORITY

### **Continuação (MQLONG)-saída**

Indicador de continuação configurado por componente..

Os valores a seguir podem ser especificados:

#### **MQZCI\_DEFAULT**

Continuação dependente do gerenciador de filas..

Para MQZ\_GET\_AUTHORITY, isso tem o mesmo efeito que MQZCI\_CONTINUE

#### **MQZCI\_CONTINUE**

Continue com o próximo componente

#### **MQZCI\_STOP**

Não continue com o próximo componente

### **CompCode (MQLONG)-saída**

Código de conclusão.

É um dos seguintes:

#### **MQCC\_OK**

Indica conclusão bem-sucedida.

#### **MQCC\_FAILED**

A chamada falhou.

### **Motivo (MQLONG)-saída**

Código de razão qualificando *CompCode*.

Se *CompCode* for MQCC\_OK:

#### **MQRC\_NONE**

(0, X'000') Nenhuma razão a relatar.

Se *CompCode* for MQCC\_FAILED:

#### **MQRC\_NOT\_AUTHORIZED**

(2035, X'7F3') Não autorizado para acesso.

#### **MQRC\_SERVICE\_ERROR**

(2289, X'8F1') Ocorreu um erro inesperado ao acessar o serviço.

#### **MQRC\_SERVICE\_NOT\_AVAILABLE**

(2285, X'8ED') Serviço subjacente não disponível.

#### **MQRC\_UNKNOWN\_ENTITY**

(2292, X'8F4') Entidade desconhecida para serviço.

Para obter mais informações sobre esses códigos de razão, consulte [Mensagens e códigos de razão](#).

## **Chamada C**

```
MQZ_GET_AUTHORITY (QMgrName, EntityName, EntityType, ObjectName,  
                  ObjectType, &Authority, ComponentData,  
                  &Continuation, &CompCode, &Reason);
```

Os parâmetros transmitidos para o serviço são declarados da seguinte forma:

```

MQCHAR48  QMgrName;           /* Queue manager name */
MQCHAR12  EntityName;       /* Entity name */
MQLONG    EntityType;       /* Entity type */
MQCHAR48  ObjectName;       /* Object name */
MQLONG    ObjectType;       /* Object type */
MQLONG    Authority;        /* Authority of entity */
MQBYTE    ComponentData[n]; /* Component data */
MQLONG    Continuation;     /* Continuation indicator set by
                             component */
MQLONG    CompCode;         /* Completion code */
MQLONG    Reason;          /* Reason code qualifying CompCode */

```

## IBM i MQZ\_GET\_EXPLICIT\_AUTHORITY (Obter autoridade explícita) em IBM i

Esta função é fornecida por um componente de serviço de autorização MQZAS\_VERSION\_1 e é chamada pelo gerenciador de filas para recuperar a autoridade que um grupo nomeado tem para acessar um objeto especificado (mas sem a autoridade adicional do grupo **nobody**) ou a autoridade que o grupo primário do proprietário nomeado tem para acessar um objeto especificado.

O identificador de função para essa função (para MQZEP) é MQZID\_GET\_EXPLICIT\_AUTHORITY

### Sintaxe

**MQZ\_GET\_EXPLICIT\_AUTHORITY** (*QMgrName, EntityName, EntityType, ObjectName, ObjectType, Authority, ComponentData, Continuation, CompCode, Reason*)

### Parâmetros

A chamada MQZ\_GET\_EXPLICIT\_AUTHORITY possui os seguintes parâmetros.

#### QMgrName (MQCHAR48)-entrada de

Nome do gerenciador de filas.

O nome do gerenciador de fila chamando o componente. Esse nome é preenchido com espaços em branco até o comprimento total do parâmetro; o nome não é terminado por um caractere nulo.

O nome do gerenciador de filas é transmitido para o componente para obter informações; a interface de serviço de autorização não requer que o componente faça uso dele de nenhuma maneira definida

#### EntityName (MQCHAR12)-entrada

Nome da entidade..

O nome da entidade da qual o acesso ao objeto deve ser recuperado. O comprimento máximo da sequência é de 12 caracteres; se ela for mais curta do que ela for preenchida à direita com espaços em branco. O nome não é terminado por um caractere nulo.

#### EntityType (MQLONG)-entrada

Tipo de entidade..

O tipo de entidade especificado por *EntityName* O valor a seguir pode ser especificado:

#### MQZAET\_PRINCIPAL

Principal.

#### MQZAET\_GROUP

.

#### ObjectName (MQCHAR48)-entrada de

Nome do objeto.

O nome do objeto para o qual a autoridade da entidade deve ser recuperada O comprimento máximo da sequência é de 48 caracteres; se for mais curto do que isso, será preenchido à direita com espaços em branco. O nome não é terminado por um caractere nulo.

Se *ObjectType* for MQOT\_Q\_MGR, esse nome será o mesmo que *QMgrName*

### **ObjectType (MQLONG)-entrada**

Tipo de objeto.

O tipo de entidade especificado por *ObjectName* É um dos seguintes:

#### **MQOT\_AUTH\_INFO**

Informações de autenticação..

#### **MQOT\_CHANNEL**

Canal.

#### **MQOT\_CLNTCONN\_CHANEXOL**

Canal de conexão do cliente.

#### **MQOT\_LISTENER**

Ouvinte.

#### **MQOT\_NAMELIST**

Lista de nomes

#### **MQOT\_PROCESS**

process definition.

#### **MQOT\_Q**

Fila.

#### **MQOT\_Q\_MGR**

Gerenciador de Filas

#### **MQOT\_SERVICE**

Serviço.

### **Autoridade (MQLONG)-saída**

Autoridade da entidade..

Se a entidade tiver uma autoridade, esse campo será igual à operação de autorização apropriada (constante MQZAO\_\*).. Se tiver mais de uma autoridade, esse campo será o OR bit a bit das constantes MQZAO\_\* correspondentes.

### **ComponentData (Comprimento de MQBYTE x ComponentData)-entrada/saída**

Dados do componente..

Esses dados são mantidos pelo gerenciador de filas em nome desse componente específico; quaisquer mudanças feitas nele por qualquer uma das funções fornecidas por esse componente são preservadas e apresentadas na próxima vez em que uma das funções desse componente for chamada de.

O comprimento dessa área de dados é passado pelo gerenciador de filas no parâmetro

**ComponentDataLength** da chamada MQZ\_INIT\_AUTHORITY

### **Continuação (MQLONG)-saída**

Indicador de continuação configurado por componente..

Os valores a seguir podem ser especificados:

#### **MQZCI\_DEFAULT**

Continuação dependente do gerenciador de filas..

Para MQZ\_GET\_EXPLICIT\_AUTHORITY, isso tem o mesmo efeito que MQZCI\_CONTINUE

#### **MQZCI\_CONTINUE**

Continue com o próximo componente

#### **MQZCI\_STOP**

Não continue com o próximo componente

### **CompCode (MQLONG)-saída**

Código de conclusão.

É um dos seguintes:

**MQCC\_OK**

Indica conclusão bem-sucedida.

**MQCC\_FAILED**

A chamada falhou.

**Motivo (MQLONG)-saída**

Código de razão qualificando *CompCode*.

Se *CompCode* for MQCC\_OK:

**MQRC\_NONE**

(0, X'000') Nenhuma razão a relatar.

Se *CompCode* for MQCC\_FAILED:

**MQRC\_NOT\_AUTHORIZED**

(2035, X'7F3') Não autorizado para acesso.

**MQRC\_SERVICE\_ERROR**

(2289, X'8F1') Ocorreu um erro inesperado ao acessar o serviço.

**MQRC\_SERVICE\_NOT\_AVAILABLE**

(2285, X'8ED') Serviço subjacente não disponível.

**MQRC\_UNKNOWN\_ENTITY**

(2292, X'8F4') Entidade desconhecida para serviço.

Para obter mais informações sobre esses códigos de razão, consulte [Mensagens e códigos de razão](#).

## Chamada C

```
MQZ_GET_EXPLICIT_AUTHORITY (QMgrName, EntityName, EntityType,  
                             ObjectName, ObjectType, &Authority,  
                             ComponentData, &Continuation,  
                             &CompCode, &Reason);
```

Os parâmetros transmitidos para o serviço são declarados da seguinte forma:

```
MQCHAR48 QMgrName;          /* Queue manager name */  
MQCHAR12 EntityName;       /* Entity name */  
MQLONG   EntityType;       /* Entity type */  
MQCHAR48 ObjectName;       /* Object name */  
MQLONG   ObjectType;       /* Object type */  
MQLONG   Authority;        /* Authority of entity */  
MQBYTE   ComponentData[n]; /* Component data */  
MQLONG   Continuation;     /* Continuation indicator set by  
                             component */  
MQLONG   CompCode;         /* Completion code */  
MQLONG   Reason;          /* Reason code qualifying CompCode */
```



## **MQZ\_INIT\_AUTHORITY (Inicializar serviço de autorização) no IBM i**

Essa função é fornecida por um componente de serviço de autorização e é chamada pelo gerenciador de filas durante a configuração do componente. É esperado chamar MQZEP para fornecer informações para o gerenciador de filas.

O identificador de função para esta função (para MQZEP) é MQZID\_INIT\_AUTHORITY.

### Sintaxe

**MQZ\_INIT\_AUTHORITY (Hconfig, Options, QMgrName, ComponentDataLength,  
ComponentData, Version, CompCode, Reason)**



## Parâmetros

A chamada MQZ\_INIT\_AUTHORITY tem os seguintes parâmetros.

### Hconfig (MQHCONFIG)-entrada

Identificador de configuração

Essa manipulação representa o componente específico sendo inicializado. Ele deve ser usado pelo componente ao chamar o gerenciador de filas com a função MQZEP

### Opções (MQLONG)-entrada

Opções de inicialização..

É um dos seguintes:

#### **MQZIO\_PRIMARY**

Inicialização primária..

#### **MQZIO\_SECONDARY**

Inicialização secundária..

### QMgrName (MQCHAR48)-entrada de

Nome do gerenciador de filas.

O nome do gerenciador de fila chamando o componente. Esse nome é preenchido com espaços em branco até o comprimento total do parâmetro; o nome não é terminado por um caractere nulo.

O nome do gerenciador de filas é transmitido para o componente para obter informações; a interface de serviço de autorização não requer que o componente faça uso dele de nenhuma maneira definida

### Comprimento de ComponentData(MQLONG)-entrada

Comprimento dos dados do componente

Comprimento em bytes da área *ComponentData* . Esse comprimento é definido no componente de dados de configuração.

### ComponentData (Comprimento de MQBYTE x ComponentData)-entrada/saída

Dados do componente..

Isso é inicializado para todos os zeros antes de chamar a função de inicialização primária do componente.. Esses dados são mantidos pelo gerenciador de filas em nome desse componente específico; quaisquer mudanças feitas nele por qualquer uma das funções (incluindo a função de inicialização) fornecidas por esse componente são preservadas e apresentadas na próxima vez que uma das funções desse componente for chamada.

### Versão (MQLONG)-entrada/saída

Número da versão..

Na entrada para a função de inicialização, identifica o número da versão *mais alto* que o gerenciador de filas suporta. A função de inicialização deve alterar isso, se necessário, para a versão da interface que *ele* suporta.. Se no retorno o gerenciador de filas não suportar a versão retornada pelo componente, ele chamará a função MQZ\_TERM\_AUTHORITY do componente e não fará uso adicional desse componente.

Os seguintes valores são suportados:

#### **MQZAS\_VERSION\_1**

Versão 1.

#### **MQZAS\_VERSION\_2**

Versão 2.

#### **MQZAS\_VERSION\_3**

Versão 3.

#### **MQZAS\_VERSION\_4**

Versão 4.

**MQZAS\_VERSION\_5**

Versão 5.

**MQZAS\_VERSION\_6**

IBM WebSphere MQ 6.

**CompCode (MQLONG)-saída**

Código de conclusão.

É um dos seguintes:

**MQCC\_OK**

Indica conclusão bem-sucedida.

**MQCC\_FAILED**

A chamada falhou.

**Motivo (MQLONG)-saída**

Código de razão qualificando *CompCode*.

Se *CompCode* for MQCC\_OK:

**MQRC\_NONE**

(0, X'000') Nenhuma razão a relatar.

Se *CompCode* for MQCC\_FAILED:

**MQRC\_INITIALIZATION\_FAILED**

(2286, X'8EE') A inicialização falhou por uma razão indefinida..

**MQRC\_SERVICE\_NOT\_AVAILABLE**

(2285, X'8ED') Serviço subjacente não disponível.

Para obter mais informações sobre esses códigos de razão, consulte [Mensagens e códigos de razão](#).

**Chamada C**

```
MQZ_INIT_AUTHORITY (Hconfig, Options, QMgrName, ComponentDataLength,
                    ComponentData, &Version, &CompCode,
                    &Reason);
```

Os parâmetros transmitidos para o serviço são declarados da seguinte forma:

```
MQHCONFIG  Hconfig;           /* Configuration handle */
MQLONG     Options;           /* Initialization options */
MQCHAR48   QMgrName;         /* Queue manager name */
MQLONG     ComponentDataLength; /* Length of component data */
MQBYTE     ComponentData[n]; /* Component data */
MQLONG     Version;          /* Version number */
MQLONG     CompCode;         /* Completion code */
MQLONG     Reason;           /* Reason code qualifying CompCode */
```

**MQZ\_INQUIRE (Consultar serviço de autorização) em IBM i**

Esta função é fornecida por um componente de serviço de autorização MQZAS\_VERSION\_5 e é chamada pelo gerenciador de filas para consultar a funcionalidade suportada.. Quando diversos componentes de serviço são usados, os componentes de serviço são chamados em ordem reversa para a ordem em que eles foram instalados.

O identificador de função para esta função (para MQZEP) é MQZID\_INQUIRE.

**Sintaxe****MQZ\_INQUIRE**

(*QMgrName*, *SelectorCount*, *Selectors*, *IntAttrCount*, *IntAttrs*, *CharAttrLength*, *CharAttrs*, *SelectorReturned*, *ComponentData*, *Continuation*, *CompCode*, *Reason*)

## Parâmetros

A chamada MQZ\_INQUIRE possui os seguintes parâmetros.

### **QMgrName (MQCHAR48)-entrada de**

Nome do gerenciador de filas.

O nome do gerenciador de fila chamando o componente. Esse nome é preenchido com espaços em branco até o comprimento total do parâmetro; o nome não é terminado por um caractere nulo.

O nome do gerenciador de filas é transmitido para o componente para obter informações; a interface de serviço de autorização não requer que o componente faça uso dele de nenhuma maneira definida

### **SelectorCount (MQLONG)-entrada**

Número de seletores..

O número de seletores fornecidos no parâmetro Selectors

O valor deve estar entre zero e 256.

### **Seletores (MQLONG x SelectorCount)-entrada**

Seletores.

Matriz de seletores.. Cada seletor identifica um atributo necessário e deve ser de um dos seguintes tipos:

- MQIACF\_ \* (número inteiro)
- MQCACF\_ \* (caractere)

Os seletores podem ser especificados em qualquer ordem O número de seletores na matriz é indicado pelo parâmetro SelectorCount .

Os atributos de número inteiro identificados por seletores são retornados no parâmetro IntAttrs na mesma ordem que aparecem em Seletores.

Os atributos de caractere identificados por seletores são retornados no parâmetro CharAttrs na mesma ordem em que eles aparecem em Seletores

### **IntAttrContagem (MQLONG)-entrada**

Número de atributos de número inteiro.

O número de atributos inteiros fornecidos no parâmetro IntAttrs .

O valor deve estar no intervalo de 0 a 256.

### **IntAttrs (MQLONG x IntAttrContagem)-saída**

Atributos de número inteiro..

Matriz de atributos de número inteiro Os atributos de número inteiro são retornados na mesma ordem que os seletores de número inteiro correspondentes na matriz Selectors.

### **CharAttrContagem (MQLONG)-entrada**

Comprimento do buffer de atributo de caractere

O comprimento em bytes do parâmetro CharAttrs .

O valor deve pelo menos somar os comprimentos dos atributos de caracteres solicitados. Se nenhum atributo de caractere for solicitado, zero será um valor válido.

### **CharAttrs (MQLONG x CharAttrContagem)-saída**

Buffer de atributos de caractere

Buffer contendo atributos de caracteres concatenados juntos. Os atributos de caractere são retornados na mesma ordem que os seletores de caracteres correspondentes na matriz de Seletores

O comprimento do buffer é fornecido pelo parâmetro de contagem CharAttr.

### **SelectorReturned (MQLONGxSelectorContagem)-entrada**

Seletor retornado.

Matriz de valores que identificam quais atributos foram retornados do conjunto solicitado pelos seletores no parâmetro *Selectors*. O número de valores nesta matriz é indicado pelo parâmetro *SelectorCount* .. Cada valor na matriz está relacionado ao seletor da posição correspondente na matriz *Selectores*. Cada valor é um dos seguintes:

**MQZSL\_RETORNADO**

O atributo solicitado pelo seletor correspondente no parâmetro *Selectors* foi retornado

**MQZSL\_NOT\_RETORNADO**

O atributo solicitado pelo seletor correspondente no parâmetro *Selectors* não foi retornado

A matriz é inicializada com todos os valores como *MQZSL\_NOT\_RETORNADO*. Quando um componente de serviço de autorização retorna um atributo, ele configura o valor apropriado na matriz para *MQZSL\_RETURNS*. Isso permite que qualquer outro componente de serviço de autorização, para o qual a chamada de consulta é feita, identifique quais atributos já foram retornados..

**ComponentData (Comprimento de MQBYTE x ComponentData)-entrada/saída**

Dados do componente..

Esses dados são mantidos pelo gerenciador de filas em nome desse componente específico; quaisquer mudanças feitas nele por qualquer uma das funções fornecidas por esse componente são preservadas e apresentadas na próxima vez em que uma das funções desse componente for chamada de.

O comprimento dessa área de dados é passado pelo gerenciador de filas no parâmetro **ComponentDataLength** da chamada *MQZ\_INIT\_AUTHORITY*

**Continuação (MQLONG)-saída**

Sinalizador de continuação

Os valores a seguir podem ser especificados:

**MQZCI\_DEFAULT**

Continuação dependente de outros componentes..

**MQZCI\_STOP**

Não continue com o próximo componente

**CompCode (MQLONG)-saída**

Código de conclusão.

É um dos seguintes:

**MQCC\_OK**

Indica conclusão bem-sucedida.

**MQCC\_WARNING**

Conclusão parcial.

**MQCC\_FAILED**

A chamada falhou.

**Motivo (MQLONG)-saída**

Código de razão qualificando *CompCode*.

Se *CompCode* for *MQCC\_OK*:

**MQRC\_NONE**

(0, X'000') Nenhuma razão a relatar.

Se *CompCode* for *MQCC\_WARNING*:

**MQRC\_CHAR\_ATTRS\_TOO\_SHORT**

Não há espaço suficiente para os atributos de caracteres

**MQRC\_INT\_COUNT\_TOO\_SMALL**

Não há espaço suficiente para atributos inteiros.

Se *CompCode* for *MQCC\_FAILED*:

**MQRC\_SELECTOR\_COUNT\_ERROR**

O número de seletores não é válido

**MQRC\_SELECTOR\_ERROR**

Seletor de atributo inválido.

**MQRC\_SELECTOR\_LIMIT\_EXCEDIDO**

Muitos seletores especificados.

**MQRC\_INT\_ATTR\_COUNT\_ERROR**

O número de atributos de número inteiro não é válido.

**MQRC\_INT\_ATTRS\_ARRAY\_ERROR**

Matriz de atributos de número inteiro inválida.

**MQRC\_CHAR\_ATTR\_LENGTH\_ERROR**

O número de atributos de caracteres não é válido

**MQRC\_CHAR\_ATTRS\_ERROR**

A cadeia de atributos de caracteres não é válida

**MQRC\_SERVICE\_ERROR**

(2289, X'8F1') Ocorreu um erro inesperado ao acessar o serviço.

**Chamada C**

```
MQZ_INQUIRE (QMgrName, SelectorCount, Selectors, IntAttrCount,
             &IntAttrs, CharAttrLength, &CharAttrs,
             SelectorReturned, ComponentData, &Continuation,
             &CompCode, &Reason);
```

Os parâmetros transmitidos para o serviço são declarados da seguinte forma:

```
MQCHAR48  QMgrName;           /* Queue manager name */
MQLONG    SelectorCount;     /* Selector count */
MQLONG    Selectors[n];      /* Selectors */
MQLONG    IntAttrCount;      /* IntAttrs count */
MQLONG    IntAttrs[n];       /* Integer attributes */
MQLONG    CharAttrCount;     /* CharAttrs count */
MQLONG    CharAttrs[n];      /* Character attributes */
MQLONG    SelectorReturned[n]; /* Selector returned */
MQBYTE    ComponentData[n];  /* Component data */
MQLONG    Continuation;      /* Continuation indicator set by
                               component */
MQLONG    CompCode;          /* Completion code */
MQLONG    Reason;           /* Reason code qualifying CompCode */
```

**MQZ\_REFRESH\_CACHE (Atualizar todas as autorizações) em IBM i**

Essa função é fornecida por um componente de serviço de autorização do MQZAS\_VERSION\_3 É chamado pelo gerenciador de filas para atualizar a lista de autorizações mantidas internamente pelo componente.

O identificador de função para essa função (para MQZEP) é MQZID\_REFRESH\_CACHE (8L).

**Sintaxe****MQZ\_REFRESH\_CACHE**

(QMgrName, ComponentData, Continuação, CompCode, Motivo)

**Parâmetros****QMgrName (MQCHAR48)-entrada**

Nome do gerenciador de filas.

O nome do gerenciador de fila chamando o componente. Esse nome é preenchido com espaços em branco até o comprimento total do parâmetro; o nome não é terminado por um caractere nulo.

O nome do gerenciador de filas é transmitido para o componente para obter informações; a interface de serviço de autorização não requer que o componente o use de nenhuma maneira definida

### **ComponentData (MQBYTE x ComponentDataComprimento) -entrada/saída**

Dados do componente..

Esses dados são mantidos pelo gerenciador de filas em nome desse componente específico. Quaisquer mudanças feitas por qualquer uma das funções fornecidas por esse componente são preservadas e apresentadas na próxima vez que uma função do componente for chamada.

O comprimento dessa área de dados é transmitido pelo gerenciador de fila no parâmetro *ComponentData* da chamada MQZ\_INIT\_AUTHORITY

### **Continuação (MQLONG)-saída.**

Indicador de continuação configurado por componente..

Os valores a seguir podem ser especificados:

#### **MQZCI\_DEFAULT**

Continuação dependente do gerenciador de filas..

Para MQZ\_REFRESH\_CACHE, isso tem o mesmo efeito que MQZCI\_CONTINUE

#### **MQZCI\_CONTINUE**

Continue com o próximo componente

#### **MQZCI\_STOP**

Não continue com o próximo componente

### **CompCode (MQLONG)-saída**

Código de conclusão.

É um dos seguintes:

#### **MQCC\_OK**

Indica conclusão bem-sucedida.

#### **MQCC\_FAILED**

A chamada falhou.

### **Motivo (MQLONG)-saída.**

Código de razão qualificando *CompCode*.

Se *CompCode* for MQCC\_OK:

#### **MQRC\_NONE**

(0, X'000') Nenhuma razão a relatar.

Se *CompCode* for MQCC\_FAILED:

#### **MQRC\_SERVICE\_ERROR**

(2289, X'8F1') Ocorreu um erro inesperado ao acessar o serviço.

## **Chamada C**

```
MQZ_REFRESH_CACHE (QMgrName, ComponentData,  
                  &Continuation, &CompCode, &Reason);
```

Declare os parâmetros conforme a seguir:

```
MQCHAR48  QMgrName;          /* Queue manager name */  
MQBYTE    ComponentData[n]; /* Component data */  
MQLONG    Continuation;     /* Continuation indicator set by  
                             component */
```

```
MQLONG   CompCode;          /* Completion code */
MQLONG   Reason;           /* Reason code qualifying CompCode */
```

## IBM i MQZ\_SET\_AUTHORITY (Configurar autoridade) em IBM i

Essa função é fornecida por um componente de serviço de autorização MQZAS\_VERSION\_1 e é chamada pelo gerenciador de filas para configurar a autoridade que uma entidade tem para acessar o objeto especificado.

O identificador de função para esta função (para MQZEP) é MQZID\_SET\_AUTHORITY

**Nota:** Esta função substitui quaisquer autoridades existentes. Para preservar quaisquer autoridades existentes, você deve configurá-las novamente com essa função.

### Sintaxe

**MQZ\_SET\_AUTHORITY** (*QMgrName, EntityName, EntityType, ObjectName, ObjectType, Authority, ComponentData, Continuation, CompCode, Reason*)

### Parâmetros

A chamada MQZ\_SET\_AUTHORITY possui os seguintes parâmetros.

#### QMgrName (MQCHAR48)-entrada de

Nome do gerenciador de filas.

O nome do gerenciador de fila chamando o componente. Esse nome é preenchido com espaços em branco até o comprimento total do parâmetro; o nome não é terminado por um caractere nulo.

O nome do gerenciador de filas é transmitido para o componente para obter informações; a interface de serviço de autorização não requer que o componente faça uso dele de nenhuma maneira definida

#### EntityName (MQCHAR12)-entrada

Nome da entidade..

O nome da entidade para a qual o acesso ao objeto deve ser definido O comprimento máximo da sequência é de 12 caracteres; se ela for mais curta do que ela for preenchida à direita com espaços em branco. O nome não é terminado por um caractere nulo.

#### EntityType (MQLONG)-entrada

Tipo de entidade..

O tipo de entidade especificado por *EntityName* O valor a seguir pode ser especificado:

**MQZAET\_PRINCIPAL**

Principal.

**MQZAET\_GROUP**

#### ObjectName (MQCHAR48)-entrada de

Nome do objeto.

O nome do objeto para o qual o acesso é necessário O comprimento máximo da sequência é de 48 caracteres; se for mais curto do que isso, será preenchido à direita com espaços em branco. O nome não é terminado por um caractere nulo.

Se *ObjectType* for MQOT\_Q\_MGR, esse nome será o mesmo que *QMgrName*

#### ObjectType (MQLONG)-entrada

Tipo de objeto.

O tipo de entidade especificado por *ObjectName* É um dos seguintes:

**MQOT\_AUTH\_INFO**

Informações de autenticação..

**MQOT\_CHANNEL**

Canal.

**MQOT\_CLNTCONN\_CHANEXOL**

Canal de conexão do cliente.

**MQOT\_LISTENER**

Ouvinte.

**MQOT\_NAMELIST**

Lista de nomes

**MQOT\_PROCESS**

process definition.

**MQOT\_Q**

Fila.

**MQOT\_Q\_MGR**

Gerenciador de Filas

**MQOT\_SERVICE**

Serviço.

**Autoridade (MQLONG)-entrada**

A autoridade a ser verificada

Se uma autorização estiver sendo configurada, esse campo será igual à operação de autorização apropriada (constante MQZAO\_\*). Se mais de uma autorização estiver sendo configurada, será a OR bit a bit das constantes MQZAO\_\* correspondentes.

**ComponentData (Comprimento de MQBYTE x ComponentData)-entrada/saída**

Dados do componente..

Esses dados são mantidos pelo gerenciador de filas em nome desse componente específico; quaisquer mudanças feitas nele por qualquer uma das funções fornecidas por esse componente são preservadas e apresentadas na próxima vez em que uma das funções desse componente for chamada de.

O comprimento dessa área de dados é passado pelo gerenciador de filas no parâmetro **ComponentDataLength** da chamada MQZ\_INIT\_AUTHORITY

**Continuação (MQLONG)-saída**

Indicador de continuação configurado por componente..

Os valores a seguir podem ser especificados:

**MQZCI\_DEFAULT**

Continuação dependente do gerenciador de filas..

Para MQZ\_SET\_AUTHORITY, isso tem o mesmo efeito que MQZCI\_STOP

**MQZCI\_CONTINUE**

Continue com o próximo componente

**MQZCI\_STOP**

Não continue com o próximo componente

**CompCode (MQLONG)-saída**

Código de conclusão.

É um dos seguintes:

**MQCC\_OK**

Indica conclusão bem-sucedida.

**MQCC\_FAILED**

A chamada falhou.

**Motivo (MQLONG)-saída**

Código de razão qualificando *CompCode*.



Se *CompCode* for MQCC\_OK:

#### **MQRC\_NONE**

(0, X'000') Nenhuma razão a relatar.

Se *CompCode* for MQCC\_FAILED:

#### **MQRC\_NOT\_AUTHORIZED**

(2035, X'7F3') Não autorizado para acesso.

#### **MQRC\_SERVICE\_ERROR**

(2289, X'8F1') Ocorreu um erro inesperado ao acessar o serviço.

#### **MQRC\_SERVICE\_NOT\_AVAILABLE**

(2285, X'8ED') Serviço subjacente não disponível.

#### **MQRC\_UNKNOWN\_ENTITY**

(2292, X'8F4') Entidade desconhecida para serviço.

## Chamada C

```
MQZ_SET_AUTHORITY (QMgrName, EntityName, EntityType, ObjectName,  
                  ObjectType, Authority, ComponentData,  
                  &Continuation, &CompCode, &Reason);
```

Os parâmetros transmitidos para o serviço são declarados da seguinte forma:

```
MQCHAR48  QMgrName;           /* Queue manager name */  
MQCHAR12  EntityName;        /* Entity name */  
MQLONG    EntityType;        /* Entity type */  
MQCHAR48  ObjectName;       /* Object name */  
MQLONG    ObjectType;       /* Object type */  
MQLONG    Authority;        /* Authority to be checked */  
MQBYTE    ComponentData[n]; /* Component data */  
MQLONG    Continuation;     /* Continuation indicator set by  
                             component */  
MQLONG    CompCode;         /* Completion code */  
MQLONG    Reason;          /* Reason code qualifying CompCode */
```

## MQZ\_TERM\_AUTHORITY-Finalizar serviço de autorização

Essa função é fornecida por um componente de serviço de autorização e é chamada pelo gerenciador de filas quando ele não requer mais os serviços desse componente. A função deve executar qualquer limpeza necessária pelo componente.

O identificador de função para essa função (para MQZEP) é MQZID\_TERM\_AUTHORITY

### Sintaxe

**MQZ\_TERM\_AUTHORITY** (*Hconfig*, *Options*, *QMgrName*, *ComponentData*,  
*CompCode*, *Reason*)

### Parâmetros

A chamada MQZ\_TERM\_AUTHORITY tem os seguintes parâmetros.

#### **Hconfig (MQHCONFIG)-entrada**

Identificador de configuração

Esse identificador representa o componente específico que está sendo finalizado

#### **Opções (MQLONG)-entrada**

Opções de finalização

É um dos seguintes:

**MQZTO\_PRIMARY**

Término primário

**MQZTO\_SECONDARY**

Finalização secundária

**QMgrName (MQCHAR48)-entrada de**

Nome do gerenciador de filas.

O nome do gerenciador de fila chamando o componente. Esse nome é preenchido com espaços em branco até o comprimento total do parâmetro; o nome não é terminado por um caractere nulo.

O nome do gerenciador de filas é transmitido para o componente para obter informações; a interface de serviço de autorização não requer que o componente faça uso dele de nenhuma maneira definida

**ComponentData (Comprimento de MQBYTE x ComponentData)-entrada/saída**

Dados do componente..

Esses dados são mantidos pelo gerenciador de filas em nome desse componente específico; quaisquer mudanças feitas nele por qualquer uma das funções fornecidas por esse componente são preservadas e apresentadas na próxima vez em que uma das funções desse componente for chamada de.

O comprimento desta área de dados é transmitido pelo gerenciador de fila no parâmetro **ComponentDataLength** na chamada MQZ\_INIT\_AUTHORITY

Quando a chamada MQZ\_TERM\_AUTHORITY tiver sido concluída, o gerenciador de filas descartará esses dados

**CompCode (MQLONG)-saída**

Código de conclusão.

É um dos seguintes:

**MQCC\_OK**

Indica conclusão bem-sucedida.

**MQCC\_FAILED**

A chamada falhou.

**Motivo (MQLONG)-saída**

Código de razão qualificando *CompCode*.

Se *CompCode* for MQCC\_OK:

**MQRC\_NONE**

(0, X'000') Nenhuma razão a relatar.

Se *CompCode* for MQCC\_FAILED:

**MQRC\_SERVICE\_NOT\_AVAILABLE**

(2285, X'8ED') Serviço subjacente não disponível.

**MQRC\_TERMINATION\_FAILED**

(2287, X'8FF') A finalização falhou por uma razão indefinida.

Para obter mais informações sobre esses códigos de razão, consulte [Mensagens e códigos de razão](#).

**Chamada C**

```
MQZ_TERM_AUTHORITY (Hconfig, Options, QMgrName, ComponentData,
                    &CompCode, &Reason);
```

Os parâmetros transmitidos para o serviço são declarados da seguinte forma:

```
MQHCONFIG  Hconfig;          /* Configuration handle */
MQLONG     Options;         /* Termination options */
```

```

MQCHAR48  QMgrName;          /* Queue manager name */
MQBYTE    ComponentData[n]; /* Component data */
MQLONG    CompCode;        /* Completion code */
MQLONG    Reason;         /* Reason code qualifying CompCode */

```

## IBM i MQZAC (contexto de aplicativos) no IBM i

Esse parâmetro especifica os dados relacionados ao aplicativo de chamada

A estrutura MQZAC é usada na chamada MQZ\_AUTHENTICATE\_USER do parâmetro **ApplicationContext**.

### Campos

#### StrucId (MQCHAR4)

Identificador de estruturação.

O valor é:

##### **MQZAC\_STRUC\_ID**

Identificador para a estrutura de contexto do aplicativo.

Para a linguagem de programação C, a constante MQZAC\_STRUC\_ID\_ARRAY também é definida; isso tem o mesmo valor que MQZAC\_STRUC\_ID, mas é uma matriz de caracteres em vez de uma sequência.

Este é um campo de entrada para o serviço

#### Versão (MQLONG)

Número de versão da estrutura.

O valor é:

##### **MQZAC\_VERSION\_1**

Estrutura do contexto de aplicativos Version-1

A constante a seguir especifica o número da versão atual:

##### **MQZAC\_CURRENT\_VERSION**

Versão atual da estrutura do contexto de aplicativos

Este é um campo de entrada para o serviço

#### ProcessId (MQPID)

Identificador do processo.

O identificador de processo do aplicativo.

#### ThreadId (MQTID)

Identificador de encadeamento.

O identificador de encadeamento do aplicativo.

#### ApplName (MQCHAR28)

Nome do aplicativo.

O nome do aplicativo.

#### UserID (MQCHAR12)

Identificador do usuário.

Para sistemas IBM i, o perfil do usuário no qual a tarefa do aplicativo foi criada. (No IBM i, quando uma troca de perfil é feita com a API QWTSETP na tarefa do aplicativo, o perfil do usuário atual é retornado)

#### ID do EffectiveUser(MQCHAR12)

Identificador de usuário efetivo

Para sistemas IBM i o perfil do usuário atual da tarefa do aplicativo.

### **Ambiente (MQLONG)**

Ambiente.

Este campo especifica o ambiente a partir do qual a chamada foi feita

Isso pode ter um dos seguintes valores:

#### **MQXE\_COMMAND\_SERVER**

Servidor de comandos

#### **MQXE\_MQSC**

O interpretador de comando `runmqsc`

#### **MQXE\_MCA**

Agente do canal de mensagens

#### **MQXE\_OTHER**

Ambiente indefinido

### **CallerType (MQLONG)**

Tipo de Responsável pela Chamada

Este campo especifica o tipo de programa que fez a chamada..

Isso pode ter um dos seguintes valores:

#### **MQXACT\_EXTERNAL**

A chamada é externa ao gerenciador de filas.

#### **MQXACT\_INTERNAL**

A chamada é interna para o gerenciador de filas

### **AuthenticationType (MQLONG)**

Tipo de autenticação.

Este campo especifica o tipo de autenticação que está sendo executada

Isso pode ter um dos seguintes valores:

#### **MQZAT\_INITIAL\_CONTEXT**

A chamada de autenticação é devido ao contexto do usuário ser inicializado. Esse valor é usado durante uma chamada `MQCONN` ou `MQCONNX`.

#### **MQZAT\_CHANGE\_CONTEXT**

A chamada de autenticação é devido ao contexto do usuário ser alterado. Esse valor é usado quando o MCA muda o contexto do usuário..

v

### **BindType (MQLONG)**

Tipo de ligação.

Este campo especifica o tipo de ligação em uso.

Isso pode ter um dos seguintes valores:

#### **MQCNO\_FASTPATH\_BINDING**

Ligação de atalho.

#### **MQCNO\_SHARED\_BINDING**

Ligação compartilhada

#### **MQCNO\_ISOLATED\_BINDING**

Ligação isolada..

## **Declaração C**

```
typedef struct tagMQZAC MQZAC;  
struct tagMQZAC {  
    MQCHAR4    StructId;           /* Structure identifier */  
    MQLONG     Version;           /* Structure version number */
```

```

MQPID      ProcessId;          /* Process identifier */
MQTID      ThreadId;       /* Thread identifier */
MQCHAR28   ApplName;       /* Application name */
MQCHAR12   UserID;         /* User identifier */
MQCHAR12   EffectiveUserID; /* Effective user identifier */
MQLONG     Environment;    /* Environment */
MQLONG     CallerType;     /* Caller type */
MQLONG     AuthenticationType; /* Authentication type */
MQLONG     BindType;       /* Bind type */
};

```

## IBM i MQZAD (Dados de autoridade) no IBM i

A estrutura MQZAD é usada na chamada MQZ\_ENUMERATE\_AUTHORITY\_DATA para dois parâmetros..

Consulte “MQZ\_ENUMERATE\_AUTHORITY\_DATA (Enumerar dados de autoridade) em IBM i” na página 1757 para obter informações adicionais sobre os parâmetros **Filter** e **AuthorityBuffer** :

- MQZAD é usado para o parâmetro **Filter** que é entrada para a chamada. Este parâmetro especifica os critérios de seleção que devem ser usados para selecionar os dados de autoridade retornados pela chamada
- MQZAD também é usado para o parâmetro **AuthorityBuffer** que é saída da chamada. Este parâmetro especifica as autorizações para uma combinação de nome do perfil, tipo do objeto e entidade

### Campos

#### StrucId (MQCHAR4)

Identificador de estruturação.

O valor é:

#### MQZAD\_STRUC\_ID

Identificador para a estrutura de dados de autoridade

Para a linguagem de programação C, a constante MQZAD\_STRUC\_ID\_ARRAY também é definida; ela possui o mesmo valor que MQZAD\_STRUC\_ID, mas é uma matriz de caracteres em vez de uma sequência.

Este é um campo de entrada para o serviço

#### Versão (MQLONG)

Número de versão da estrutura.

O valor é:

#### MQZAD\_VERSION\_1

Estrutura de dados de autoridade Version-1 .

A constante a seguir especifica o número da versão atual:

#### MQZAD\_CURRENT\_VERSION

A versão atual da estrutura de dados de autoridade

Este é um campo de entrada para o serviço

#### ProfileName (MQCHAR48)

Nome do perfil.

Para o parâmetro **Filter** , esse campo é o nome do perfil do qual os dados de autoridade são necessários. Se o nome estiver inteiramente em branco até o final do campo ou o primeiro caractere nulo, serão retornados dados de autoridade para todos os nomes de perfis.

Para o parâmetro **AuthorityBuffer** , esse campo é o nome de um perfil que corresponde aos critérios de seleção especificados

#### ObjectType (MQLONG)

Tipo de objeto.

Para o parâmetro **Filter** , esse campo é o tipo de objeto para o qual os dados de autoridade são necessários. Se o valor for MQOT\_ALL, dados de autoridade para todos os tipos de objeto serão retornados.

Para o parâmetro **AuthorityBuffer** , esse campo é o tipo de objeto ao qual o perfil identificado por **ProfileName** se aplica..

O valor é um dos seguintes; para o parâmetro **Filter** , o valor MQOT\_ALL também é válido:

**MQOT\_AUTH\_INFO**

Informações de autenticação..

**MQOT\_CHANNEL**

Canal.

**MQOT\_CLNTCONN\_CHANEXOL**

Canal de conexão do cliente.

**MQOT\_LISTENER**

Ouvinte.

**MQOT\_NAMELIST**

Lista de nomes

**MQOT\_PROCESS**

process definition.

**MQOT\_Q**

Fila.

**MQOT\_Q\_MGR**

Gerenciador de Filas

**MQOT\_SERVICE**

Serviço.

**Autoridade (MQLONG)**

Autoridade.

Para o parâmetro **Filter** , esse campo é ignorado..

Para o parâmetro **AuthorityBuffer** , esse campo representa as autorizações que a entidade tem para os objetos identificados por **ProfileName** e **ObjectType**. Se a entidade tiver apenas uma autoridade, o campo será igual ao valor de autorização apropriado (constante MQZAO\_\*). Se a entidade tiver mais de uma autoridade, o campo será OR bit a bit das constantes MQZAO\_\* correspondentes.

**EntityDataPtr (PMQZED)**

O endereço da estrutura MQZED que identifica uma entidade

Para o parâmetro **Filter** , esse campo aponta para uma estrutura MQZED que identifica a entidade da qual os dados de autoridade são necessários. Se **EntityDataPtr** for o ponteiro nulo, dados de autoridade para todas as entidades serão retornados.

Para o parâmetro **AuthorityBuffer** , esse campo aponta para uma estrutura MQZED que identifica a entidade da qual os dados de autoridade retornados vieram.

**EntityType (MQLONG)**

Tipo de entidade..

Para o parâmetro **Filter** , esse campo especifica o tipo de entidade para o qual os dados de autoridade são necessários. Se o valor for MQZAET\_NONE, dados de autoridade para todos os tipos de entidade serão retornados.

Para o parâmetro **AuthorityBuffer** , esse campo especifica o tipo da entidade identificada pelo MQZED estrutura apontada por **EntityDataPtr**.

O valor é um dos seguintes; para o parâmetro **Filter** , o valor MQZAET\_NONE também é válido:

## **MQZAET\_PRINCIPAL**

Principal.

## **MQZAET\_GROUP**

### **Opções (MQAUTHOPT)**

Opções.

Este campo especifica opções que fornecem controle sobre os perfis exibidos.

Um dos seguintes deve ser especificado:

#### **MQAUTHOPT\_NAME\_ALL\_MATCHING**

Exibe todos os perfis

#### **MQAUTHOPT\_NAME\_EXPLICIT**

Exibe perfis que possuem exatamente o mesmo nome especificado no campo **ProfileName**.

Além disso, um dos seguintes também deve ser especificado:

#### **MQAUTHOPT\_ENTITY\_SET**

Exiba todos os perfis usados para calcular a autoridade acumulativa que a entidade tem para o objeto especificado por **ProfileName**. O campo **ProfileName** não deve conter caracteres curinga.

- Se a entidade especificada for um principal, para cada membro do conjunto {entity, groups} o perfil mais aplicável que se aplica ao objeto será exibido.
- Se a entidade especificada for um grupo, o perfil mais aplicável do grupo que se aplica ao objeto será exibido.
- Se esse valor for especificado, então os valores de **ProfileName**, **ObjectType**, **EntityType** e o nome da entidade especificado na estrutura MQZED **EntityDataPtr** devem ser todos não em branco.

Se tiver especificado **MQAUTHOPT\_NAME\_ALL\_MATCHING**, também será possível especificar o seguinte:

#### **MQAUTHOPT\_ENTITY\_EXPLICIT**

Exibe perfis que possuem exatamente o mesmo nome da entidade que o nome da entidade especificado na estrutura MQZED do **EntityDataPtr**

## **Declaração C**

```
typedef struct tagMQZAD MQZAD;
struct tagMQZAD {
    MQCHAR4    StrucId;           /* Structure identifier */
    MQLONG     Version;          /* Structure version number */
    MQCHAR48   ProfileName;      /* Profile name */
    MQLONG     ObjectType;       /* Object type */
    MQLONG     Authority;        /* Authority */
    PMQZED     EntityDataPtr;    /* Address of MQZED structure identifying an
    entity */
    MQLONG     EntityType;       /* Entity type */
    MQAUTHOPT  Options;         /* Options */
};
```

## **IBM i MQZED (Descritor de entidade) no IBM i**

A estrutura MQZED é usada em várias chamadas de serviço de autorização para especificar a entidade para a qual a autorização deve ser verificada.

### **Campos**

#### **StrucId (MQCHAR4)**

Identificador de estruturação.

O valor é:

### **MQZED\_STRUC\_ID**

Identificador para a estrutura do descritor de entidade

Para a linguagem de programação C, a constante MQZED\_STRUC\_ID\_ARRAY também é definida; ela tem o mesmo valor que MQZED\_STRUC\_ID, mas é uma matriz de caracteres em vez de uma sequência.

Este é um campo de entrada para o serviço

### **Versão (MQLONG)**

Número de versão da estrutura.

O valor é:

### **MQZED\_VERSION\_1**

Version-1 estrutura do descritor de entidade.

A constante a seguir especifica o número da versão atual:

### **MQZED\_CURRENT\_VERSION**

A versão atual da estrutura do descritor de entidade

Este é um campo de entrada para o serviço

### **EntityNamePtr (PMQCHAR)**

Endereço do nome da entidade..

Este é um ponteiro para o nome da entidade cuja autorização deve ser verificada

### **EntityDomainPtr (PMQCHAR)**

Endereço do nome de domínio da entidade

Este é um ponteiro para o nome do domínio que contém a definição da entidade cuja autorização deve ser verificada

### **SecurityId (MQBYTE40)**

Identificador de segurança.

Este é o identificador de segurança cuja autorização deve ser verificada

### **CorrelationPtr (MQPTR).**

Indicador de correlação.

Isso facilita a transmissão de dados correlacionais entre a função autenticar usuário e outras funções OAM apropriadas.

## **Declaração C**

```
typedef struct tagMQZED MQZED;
struct tagMQZED {
    MQCHAR4    StrucId;           /* Structure identifier */
    MQLONG     Version;          /* Structure version number */
    PMQCHAR    EntityNamePtr;    /* Address of entity name */
    PMQCHAR    EntityDomainPtr;  /* Address of entity domain name */
    MQBYTE40   SecurityId;       /* Security identifier */
    MQPTR      CorrelationPtr;   /* Address of correlation data */
}
```

## **MQZFP (Parâmetros grátis) em IBM i**

Este parâmetro especifica os dados relacionados ao recurso a ser liberado

A estrutura MQZFP é usada na chamada MQZ\_FREE\_USER do parâmetro **FreeParms** .



## Campos

### StrucId (MQCHAR4)

Identificador de estruturação.

O valor é:

#### MQZFP\_STRUC\_ID

Identificador para estrutura de parâmetros livres.

Para a linguagem de programação C, a constante MQZFP\_STRUC\_ID\_ARRAY também é definida; isso tem o mesmo valor que MQZFP\_STRUC\_ID, mas é uma matriz de caracteres em vez de uma sequência.

Este é um campo de entrada para o serviço

### Versão (MQLONG)

Número de versão da estrutura.

O valor é:

#### MQZFP\_VERSION\_1

Version-1 estrutura de parâmetros livres.

A constante a seguir especifica o número da versão atual:

#### MQZFP\_CURRENT\_VERSION

A versão atual da estrutura de parâmetros livres

Este é um campo de entrada para o serviço

### Reservado (MQBYTE8)

Campo reservado..

O valor inicial é nulo.

### CorrelationPtr (MQPTR).

Indicador de correlação.

Endereço de dados de correlações relacionados ao recurso a ser liberado

## Declaração C

```
typedef struct tagMQZFP MQZFP;
struct tagMQZFP {
    MQCHAR4    StrucId;           /* Structure identifier */
    MQLONG     Version;         /* Structure version number */
    MQBYTE8    Reserved;        /* Reserved field */
    MQPTR      CorrelationPtr;  /* Address of correlation data */
};
```

## IBM i MQZIC (contexto de identidade) no IBM i

A estrutura MQZIC é usada na chamada MQZ\_AUTHENTICATE\_USER do parâmetro **IdentityContext**.

A estrutura MQZIC contém informações de contexto de identidade, que identifica o usuário do aplicativo que primeiro colocou a mensagem em uma fila:

- O gerenciador de filas preenche o campo UserIdentifier com um nome que identifica o usuário, a maneira como o gerenciador de filas pode fazer isso depende do ambiente no qual o aplicativo está em execução.
- O gerenciador de filas preenche o campo AccountingToken com um token ou número determinado a partir do aplicativo que colocou a mensagem.
- Os aplicativos podem usar o campo ApplIdentityData para quaisquer informações extras que eles desejam incluir sobre o usuário (por exemplo, uma senha criptografada).

Os aplicativos adequadamente autorizados podem configurar o contexto de identidade usando a função MQZ\_AUTHENTICATE\_USER

Um identificador de segurança do sistema (SID) Windows é armazenado no campo AccountingToken quando uma mensagem é criada em IBM MQ for Windows. O SID pode ser usado para complementar o campo UserIdentifier para estabelecer as credenciais de um usuário.

## Campos

### StrucId (MQCHAR4)

Identificador de estruturação.

O valor é:

#### MQZIC\_STRUC\_ID

Identificador para a estrutura de contexto de identidade

Para a linguagem de programação C, a constante MQZIC\_STRUC\_ID\_ARRAY também é definida; isso tem o mesmo valor que MQZIC\_STRUC\_ID, mas é uma matriz de caracteres em vez de uma sequência.

Este é um campo de entrada para o serviço

### Versão (MQLONG)

Número de versão da estrutura.

O valor é:

#### MQZIC\_VERSION\_1

Version-1 estrutura de contexto de identidade.

A constante a seguir especifica o número da versão atual:

#### MQZIC\_CURRENT\_VERSION

A versão atual da estrutura do contexto de identidade

Este é um campo de entrada para o serviço

### UserIdentifier (MQCHAR12)

Identificador do usuário.

Isso faz parte do **contexto de identidade** da mensagem

*UserIdentifier* especifica o identificador de usuário do aplicativo que originou a mensagem. O gerenciador de filas trata essas informações como dados de caractere, mas não define o formato deles. Para obter mais informações sobre o campo *UserIdentifier*, consulte [“UserIdentifier \(MQCHAR12\) para MQMD”](#) na página 473

### AccountingToken (MQBYTE32)

Token de Contabilidade.

Isso faz parte do **contexto de identidade** da mensagem

*AccountingToken* permite que um aplicativo faça com que o trabalho feito como resultado da mensagem seja adequadamente cobrado. O gerenciador de filas trata estas informações como uma sequência de bits e não verifica seu conteúdo. Para obter mais informações sobre o campo *AccountingToken*, consulte [“AccountingToken \(MQBYTE32\) para MQMD”](#) na página 475

### ApplIdentityDados (MQCHAR32)

Dados de aplicativos relacionados à identidade.

Isso faz parte do **contexto de identidade** da mensagem

*ApplIdentityData* são informações definidas pelo conjunto de aplicativos que podem ser usadas para fornecer informações adicionais sobre a origem da mensagem. Por exemplo, ele poderia ser configurado por aplicativos em execução com autoridade de usuário adequada para indicar se os dados de identificação são confiáveis.. Para obter mais informações sobre o campo *ApplIdentityData*, consulte [“Dados de ApplIdentity\(MQCHAR32\) para MQMD”](#) na página 477

## Declaração C

```
typedef struct tagMQZED MQZED;
struct tagMQZED {
    MQCHAR4   StrucId;           /* Structure identifier */
    MQLONG    Version;          /* Structure version number */
    MQCHAR12  UserIdentifier;    /* User identifier */
    MQBYTE32  AccountingToken;  /* Accounting token */
    MQCHAR32  ApplIdentityData; /* Application data relating to identity */
};
```

## As classes e interfaces do IBM MQ .NET

As classes e interfaces do IBM MQ .NET são listadas alfabeticamente. As propriedades, métodos e construtores são descritos.

### Classe MQAsyncStatus.NET

Use MQAsyncStatus para consultar o status da atividade MQI anterior; por exemplo, consultar o sucesso de operações put assíncronas anteriores. MQAsyncStatus encapsula recursos da estrutura de dados do MQSTS.

#### Parâmetro class

```
System.Object
├── IBM.WMQ.MQBase
│   └── IBM.WMQ.MQBaseObject
│       └── IBM.WMQ.MQAsyncStatus
```

```
public class IBM.WMQ.MQAsyncStatus extends IBM.WMQ.MQBaseObject;
```

- [“Propriedades” na página 1783](#)
- [“Construtores” na página 1784](#)

#### Propriedades

Teste para MQException ser lançado ao obter propriedades.

```
public static int CompCode {get;}
```

O código de conclusão do primeiro erro ou aviso.

```
public static int Reason {get;}
```

O código de razão do primeiro erro ou aviso.

```
public static int PutSuccessCount {get;}
```

O número de chamadas put de MQI assíncronas bem-sucedidas.

```
public static int PutWarningCount {get;}
```

O número de chamadas put de MQI assíncronas que foram bem-sucedidas com um aviso.

```
public static int PutFailureCount {get;}
```

O número de chamadas put de MQI assíncronas que falharam.

```
public static int ObjectType {get;}
```

O tipo de objeto para o primeiro erro. Os valores a seguir são possíveis:

- MQC.MQOT\_ALIAS\_Q
- MQC.MQOT\_LOCAL\_Q
- MQC.MQOT\_MODEL\_Q
- MQC.MQOT\_Q
- MQC.MQOT\_REMOTE\_Q
- MQC.MQOT\_TOPIC
- 0, significando que nenhum objeto é retornado

**public static string ObjectName {get;}**

O nome do objeto.

**public static string ObjectQMgrName {get;}**

O nome do gerenciador de filas de objeto.

**public static string ResolvedObjectName {get;}**

O nome do objeto resolvido.

**public static string ResolvedObjectQMgrName {get;}**

O nome do gerenciador de filas do objeto resolvido.

## Construtores

**public MQAsyncStatus() throws MQException;**

Método do construtor, constrói um objeto com campos inicializados com zero ou em branco conforme apropriado.

## Classe MQAuthenticationInformationRecord.NET

Use MQAuthenticationInformationRecord para especificar informações sobre um autenticador que deve ser usado em uma conexão do cliente TLS IBM MQ. MQAuthenticationInformationRecord contém um registro de informações sobre autenticação, MQAIR.

### Parâmetro class

```
System.Object
└─ IBM.WMQ.MQAuthenticationInformationRecord
```

**public class IBM.WMQ.MQAuthenticationInformationRecord extends System.Object;**

- [“Propriedades” na página 1784](#)
- [“Construtores” na página 1785](#)

### Propriedades

Teste para MQException ser lançado ao obter propriedades.

**public long Version {get; set;}**

Número de versão da estrutura.

**public long AuthInfoType {get; set;}**

O tipo de informações sobre autenticação. Este atributo deve ser configurado como um dos seguintes valores:

- OCSP - A verificação do status de revogação de certificado é feita usando OCSP.

- CRLLDAP - A verificação do status de revogação de certificado é feita usando Listas de Revogação de Certificado em servidores LDAP.

**public string AuthInfoConnName {get; set;}**

O nome DNS ou endereço IP do host no qual o servidor LDAP está em execução, com um número da porta opcional. Esta palavra-chave é obrigatório.

**public string LDAPPassword {get; set;}**

A senha associada ao nome distinto do usuário que está acessando o servidor LDAP. Esta propriedade se aplica apenas quando **AuthInfoType** é configurado como CRLLDAP.

**public string LDAPUserName {get; set;}**

O nome distinto do usuário que está acessando o servidor LDAP. Quando você configura esta propriedade, LDAPUserNameLength e LDAPUserNamePtr são automaticamente configurados de modo correto. Esta propriedade se aplica apenas quando AuthInfoType é configurado como CRLLDAP.

**public string OCSPResponderURL {get; set;}**

A URL na qual o replicador OCSP pode ser contatado. Esta propriedade se aplica apenas quando AuthInfoType é configurado como OCSP

Este campo faz distinção entre maiúsculas e minúsculas Ele deve iniciar com a cadeia http:// em minúsculas. O restante da URL pode fazer distinção entre maiúsculas e minúsculas, dependendo da implementação do servidor OCSP

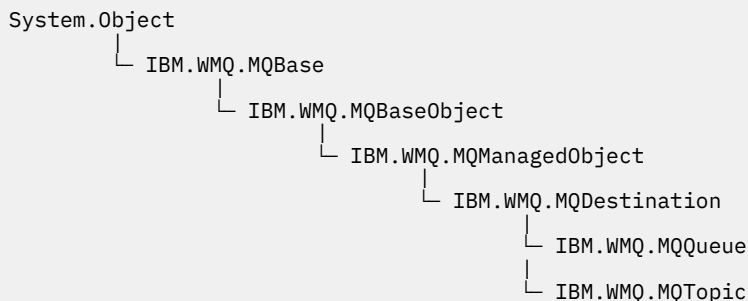
## Construtores

**MQAuthenticationInformationRecord();**

## Classe MQDestination.NET

Use MQDestination para acessar métodos que são comuns para MQQueue e MQTopic. MQDestination é uma classe-base abstrata e não pode ser instanciado

### Parâmetro class



```
public class IBM.WMQ.MQDestination extends IBM.WMQ.MQManagedObject;
```

- [“Propriedades” na página 1785](#)
- [“Métodos” na página 1786](#)
- [“Construtores” na página 1787](#)

## Propriedades

Teste para MQException ser lançado ao obter propriedades.

```
public DateTime CreationDateTime {get;}
```

A data e hora em que a fila ou tópicos foi criado. Originalmente contida em MQQueue, essa propriedade foi movida para a classe MQDestination base

Não existe valor-padrão.

```
public int DestinationType {get;}
```

Valor de número inteiro que descreve o tipo de destino que está sendo usado. Inicializado a partir do construtor de subclasses, MQQueue ou MQTopic, este valor pode ter um destes valores:

- MQOT\_Q
- MQOT\_TOPIC

Não existe valor-padrão.

## Métodos

```
public void Get(MQMessage message);
```

```
public void Get(MQMessage message, MQGetMessageOptions getMessageOptions);
```

```
public void Get(MQMessage message, MQGetMessageOptions getMessageOptions, int MaxMsgSize);
```

Lança MQException..

Obtém uma mensagem de uma fila se o destino for um objeto MQQueue ou de um tópico se o destino for um objeto MQTopic, usando uma instância padrão de MQGetMessageOptions para executar a obtenção

Se a obtenção falhar, o objeto MQMessage permanecerá inalterado. Se ele for bem-sucedido, as partes do descritor de mensagens e de dados da mensagem do MQMessage serão substituídas pelo descritor de mensagens e pelos dados da mensagem recebida.

Todas as chamadas para IBM MQ de um determinado MQQueueManager são síncronas. Portanto, se você executar um get com wait, todos os outros encadeamentos usando o mesmo MQQueueManager serão bloqueados de fazer chamadas IBM MQ adicionais até que a chamada Get seja realizada..

Se você precisar de vários encadeamentos para acessar o IBM MQ simultaneamente, cada encadeamento deverá criar seu próprio objeto MQQueueManager

### mensagem

Contém o descritor de mensagens e os dados da mensagem retornados. Alguns dos campos no descritor de mensagem são parâmetros de entrada. É importante assegurar que os parâmetros de entrada MessageId e CorrelationId sejam configurados conforme necessário.

Um cliente reconectável retorna o código de razão MQRC\_BACKED\_OUT após a reconexão bem-sucedida para mensagens recebidas em MQGM\_SYNCPOINT.

### Opções getMessage

Opções que controlam a ação de obtenção.

Usar a opção MQC.MQGM\_CONVERT pode resultar em uma exceção com o código de razão MQC.MQRC\_CONVERTED\_STRING\_TOO\_BIG ao converter de códigos de caracteres de byte único para códigos de byte duplo. Neste caso, a mensagem é copiada para o buffer sem conversão.

Se *getMessageOptions* não for especificado, a opção de mensagem usada será MQGM\_NOWAIT.

Se você usar a opção MQGM\_LOGICAL\_ORDER em um cliente reconectável, o código de razão MQRC\_RECONNECT\_INCOMPATIBLE será retornado.

### MaxMsgTamanho

A maior mensagem que este objeto de mensagem deve receber. Se a mensagem na fila for maior que este tamanho, uma de duas coisas ocorrerá:

- Se a sinalização MQGM\_ACCEPT\_TRUNCATED\_MSG for configurada no objeto MQGetMessageOptions, a mensagem será preenchida com o máximo de dados da mensagem.

possível. Uma exceção é lançada com o código de conclusão MQCC\_WARNING e o código de razão MQRC\_TRUNCATED\_MSG\_ACCEPTED

- Se o sinalizador MQGMO\_ACCEPT\_TRUNCATED\_MSG não for configurado, a mensagem permanecerá na fila. Uma exceção é lançada com o código de conclusão MQCC\_WARNING e o código de razão MQRC\_TRUNCATED\_MSG\_FAILED

Se *MaxMsgSize* não for especificado, a mensagem inteira será recuperada

```
public void Put(MQMessage message);  
public void Put(MQMessage message, MQPutMessageOptions putMessageOptions);
```

Lança MQException..

Coloca uma mensagem em uma fila se o destino for um objeto MQQueue ou publica uma mensagem em um tópico se o destino for um objeto MQTopic .

Modificações no objeto MQMessage após a chamada Put ter sido realizada não afetam a mensagem real na fila IBM MQ ou tópico de publicação.

Put atualiza as propriedades MessageId e CorrelationId do objeto MQMessage e não limpa dados da mensagem. Chamadas Put ou Get adicionais se referem às informações atualizadas no objeto MQMessage . Por exemplo, no fragmento de código a seguir, a primeira mensagem contém a e a segunda ab.

```
msg.WriteString("a");  
q.Put(msg,pmo);  
msg.WriteString("b");  
q.Put(msg,pmo);
```

### **mensagem**

Um objeto MQMessage que contém os dados do descritor de mensagens e a mensagem a ser enviada. O descritor de mensagens pode ser alterado como uma consequência deste método. Os valores no descritor de mensagens imediatamente após a conclusão desse método são os valores que foram colocados na fila ou publicados no tópico.

Os seguintes códigos de razão são retornados para um cliente reconectável:

- MQRC\_CALL\_INTERRUPTED se a conexão for interrompida ao executar uma chamada Put em uma mensagem persistente e a reconexão for bem-sucedida.
- MQRC\_NONE se a conexão for bem-sucedida ao executar uma chamada Put em uma mensagem não persistente (consulte [Recuperação de Aplicativo](#)).

### **putMessageOptions**

Opções que controlam a ação da entrada.

Se *putMessageOptions* não for especificado, a instância padrão de MQPutMessageOptions será usada.

Se você usar a opção MQPMO\_LOGICAL\_ORDER em um cliente reconectável, o código de razão MQRC\_RECONNECT\_INCOMPATIBLE será retornado.

**Nota:** Para simplicidade e desempenho, se você desejar colocar uma única mensagem em uma fila, use o objeto MQQueueManager . Put Deve-se ter um objeto MQQueue para isso.

### **Construtores**

MQDestination é uma classe-base abstrata e não pode ser instanciada. Acesse destinos usando construtores MQQueue e MQTopic ou usando MQQueueManager . AccessQueue e MQQueueManager.AccessTopic methods.

## Classe MQEnvironment.NET

Use MQEnvironment para controlar como o construtor MQQueueManager é chamado e selecionar uma conexão IBM MQ MQI client . A classe MQEnvironment contém propriedades que controlam o comportamento do IBM MQ..

### Parâmetro class

```
System.Object
└─ IBM.WMQ.MQEnvironment
```

```
public class IBM.WMQ.MQEnvironment extends System.Object;
```

- [“Propriedades-somente cliente” na página 1788](#)
- [“Propriedades” na página 1789](#)
- [“Construtores” na página 1790](#)

### Propriedades-somente cliente

Teste para MQException ser lançado ao obter propriedades.

```
public static int CertificateValPolicy {get; set;}
```

Configure qual política de validação de certificado TLS é usada para validar certificados digitais recebidos de sistemas parceiros remotos. Os valores válidos são:

- MQC.CERTIFICATE\_VALIDATION\_POLICY\_ANY
- MQC.CERTIFICATE\_VALIDATION\_POLICY\_RFC5280

```
public static ArrayList EncryptionPolicySuiteB {get; set;}
```

Configure o nível de criptografia compatível com Suite B. Os valores válidos são:

- MQC.MQ\_SUITE\_B\_NONE -Este é o valor padrão.
- MQC.MQ\_SUITE\_B\_128\_BIT
- MQC.MQ\_SUITE\_B\_192\_BIT

```
public static string Channel {get; set;}
```

O nome do canal para conectar ao gerenciador de filas de destino. Deve-se configurar a propriedade do canal antes de instanciar uma instância do MQQueueManager no modo cliente.

```
public static int FipsRequired {get; set;}
```

Especifique MQC.MQSSL\_FIPS\_YES para usar apenas algoritmos certificados por FIPS se a criptografia for executada em IBM MQ O padrão é MQC.MQSSL\_FIPS\_NO.

Se o hardware criptográfico estiver configurado, os módulos criptográficos usados serão aqueles fornecidos pelo produto de hardware Dependendo do hardware em uso, elas podem não ser certificadas por FIPS para um nível específico

```
public static string Hostname {get; set;}
```

O nome do host TCP/IP do computador no qual o servidor IBM MQ reside.. Se o nome do host não estiver configurado e nenhuma propriedade de substituição estiver configurada, o modo de ligações do servidor será usado para conectar-se ao gerenciador de filas locais.

```
public static int Port {get; set;}
```

A porta na qual se conectar. Esta é a porta na qual o servidor IBM MQ está atendendo as solicitações de conexão recebidas O valor padrão é 1414.



**public static string SSLCipherSpec {get; set;}**

Configure SSLCipherSpec para o valor do CipherSpec configurado no canal SVRCONN para ativar TLS para a conexão. O padrão é Nulo e o TLS não está ativado para a conexão

**public static string sslPeerName {get; set;}**

Um padrão de nome distinto. Se sslCipherSpec estiver configurado, essa variável poderá ser usada para assegurar que o gerenciador de filas correto seja usado.. Se configurado como nulo (padrão), o DN do gerenciador de filas não será executado. sslPeerName será ignorado se sslCipherSpec for nulo.

**V 9.3.0** **V 9.3.0** **public static string SSLKeyRepository {get; set;}**

O texto simples ou a passphrase criptografada para acessar o repositório de chaves. As passphrases do repositório de chave são criptografados para uso pelos aplicativos clientes usando o utilitário **runmqicred**

Se SSLKeyRepositoryPassword for configurado como nulo (padrão), o valor da variável de ambiente **MQKEYRPWD** ou o atributo **SSLKeyRepositoryPassword** no arquivo de configuração do cliente será usado.

**V 9.3.0** **V 9.3.0** **public static string InitialKey {get; set;}**

A chave inicial que foi usada para criptografar a passphrase do repositório de chaves especificada no SSLKeyRepositoryPassword

A chave inicial deve ser especificada se um arquivo de chave inicial foi especificado quando a passphrase do repositório de chaves foi criptografada usando o utilitário **runmqicred**.

## Propriedades

Teste para MQException ser lançado ao obter propriedades.

**public static ArrayList HdrCompList {get; set;}**

Lista de Compactação de Dados do Cabeçalho

**public static int KeyResetCount {get; set;}**

Indica o número de bytes não criptografados enviados e recebidos dentro de uma conversa TLS antes que a chave secreta seja renegociada

**public static ArrayList MQAIRArray {get; set;}**

Uma matriz de objetos do MQAuthenticationInformationRecord

**public static ArrayList MsgCompList {get; set;}**

Lista de Compactação de Dados de Mensagem

**public static string Password {get; set;}**

A senha a ser autenticada. A senha referenciada na estrutura MQCSP é preenchida configurando essa propriedade Senha.

**public static string ReceiveExit {get; set;}**

Uma saída de recebimento permite que você examine e altere dados recebidos de um gerenciador de filas. Ela é normalmente usada com uma saída de envio correspondente no gerenciador de filas. Se ReceiveExit for configurado como nulo, nenhuma saída de recebimento será chamada.

**public static string ReceiveUserData {get; set;}**

Os dados do usuário associados a uma saída de recebimento. Limitados a 32 caracteres.

**public static string SecurityExit {get; set;}**

Uma saída de segurança permite que você customize os fluxos de segurança que ocorrem quando uma tentativa de conectar-se a um gerenciador de filas é feita. Se SecurityExit for configurado para nulo, nenhuma saída de segurança será chamada.

**public static string SecurityUserData {get; set;}**

Os dados do usuário associado a uma saída de segurança. Limitados a 32 caracteres.

**public static string SendExit {get; set;}**

Uma saída de envio permite examinar ou alterar os dados enviados para um gerenciador de filas. Ela é normalmente usada com uma saída de recebimento correspondente no gerenciador de filas. Se o SendExit for configurado como nulo, nenhuma saída de envio será chamada

**public static string SendUserData {get; set;}**

Os dados do usuário associados a uma saída de envio. Limitados a 32 caracteres.

**public static string SharingConversations {get; set;}**

O campo SharingConversations é usado em conexões de aplicativos .NET , quando esses aplicativos não estão usando uma tabela de definições de canal do cliente (CCDT).

SharingConversations determina o número máximo de conversações que podem ser compartilhadas em um soquete associado a esta conexão

Um valor de 0 significa que o canal opera como antes do IBM WebSphere MQ 7.0, com relação ao compartilhamento de conversa, leitura antecipada e pulsação.

O campo é transmitido na tabela hash de propriedades como um SHARING\_CONVERSATIONS\_PROPERTY, ao instanciar um gerenciador de filas do IBM MQ

Se você não especificar SharingConversations, um valor padrão 10 será usado.

**public static string SSLCryptoHardware {get; set;}**

Configura o nome da cadeia de parâmetros necessária para configurar o hardware de criptografia presente no sistema. SSLCryptoHardware será ignorado se sslCipherSpec for nulo.

**public static string SSLKeyRepository {get; set;}**

Configure o nome completo do arquivo completo do repositório de chaves

 Se a extensão do arquivo não for fornecida ele será assumido como .kdb

Se SSLKeyRepository for configurado como nulo (padrão), a variável de ambiente MQSSLKEYR do certificado será usada para localizar o repositório de chaves.

**public static string UserId {get; set;}**

O ID do usuário a ser autenticado. O ID do usuário referenciado a partir da estrutura MQCSP é preenchido configurando UserId. Autentique o UserId usando uma saída de API ou de Segurança

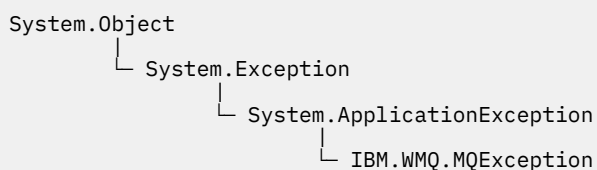
## Construtores

**public MQEnvironment()**

## Classe MQException.NET

Use MQException para descobrir a conclusão e o código de razão de uma função IBM MQ com falha.. Um MQException é lançado sempre que ocorre um erro de IBM MQ

### Parâmetro class



**public class IBM.WMQ.MQException extends System.ApplicationException;**

- [“Propriedades” na página 1791](#)

- [“Construtores” na página 1791](#)

## Propriedades

**public int CompletionCode {get; set;}**

O código de conclusão IBM MQ associado ao erro. Os valores possíveis são:

- MQException.MQCC\_OK
- MQException.MQCC\_WARNING
- MQException.MQCC\_FAILED

**public int ReasonCode {get; set;}**

O código de razão IBM MQ que descreve o erro

## Construtores

**public MQException(int completionCode, int reasonCode)**

**completionCode**

O código de conclusão do IBM MQ

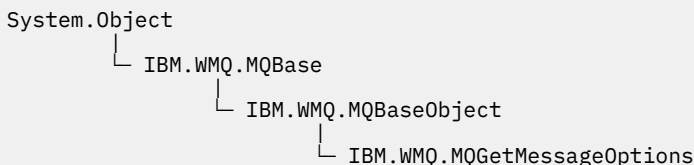
**reasonCode**

O código de conclusão do IBM MQ

## Classe MQGetMessageOptions.NET

Use MQGetMessageOptions para especificar como as mensagens são recuperados Ele modifica o comportamento do MQDestination.Get

### Parâmetro class



```
public class IBM.WMQ.MQGetMessageOptions extends IBM.WMQ.MQBaseObject;
```

- [“Propriedades” na página 1791](#)
- [“Construtores” na página 1794](#)

## Propriedades

**Nota:** O comportamento de algumas das opções disponíveis nesta classe depende do ambiente no qual elas são usadas. Esses elementos são marcados com um asterisco \*.

Teste para MQException ser lançado ao obter propriedades.

**public int GroupStatus {get;}\***

GroupStatus indica se a mensagem recuperada está em um grupo e se ela é a última no grupo Os valores possíveis são:

**MQC.MQGS\_LAST\_MSG\_IN\_GROUP**

Mensagem é a última ou única mensagem no grupo.

**MQC.MQGS\_MSG\_IN\_GROUP**

A mensagem está em um grupo, mas não é a última no grupo.

**MQC.MQGS\_NOT\_IN\_GROUP**

A mensagem não está em um grupo.

**public int MatchOptions {get; set;}\***

MatchOptions determina como uma mensagem é selecionada.. As seguintes opções de correspondência podem ser configuradas:

**MQC.MQMO\_MATCH\_CORREL\_ID**

ID de correlação a ser correspondido.

**MQC.MQMO\_MATCH\_GROUP\_ID**

ID do grupo a ser correspondido.

**MQC.MQMO\_MATCH\_MSG\_ID**

ID de mensagem a ser correspondido.

**MQC.MQMO\_MATCH\_MSG\_SEQ\_NUMBER**

Número de sequência de mensagem de correspondência.

**MQC.MQMO\_NONE**

Nenhuma correspondência necessária.

**public int Options {get; set;}**

Opções controlam a ação de MQQueue . get.. Qualquer um dos valores a seguir pode ser especificado Se mais de uma opção for necessária, os valores poderão ser incluídos ou combinados usando o operador OR bit a bit.

**MQC.MQGMO\_ACCEPT\_TRUNCATED\_MSG**

Permitir truncamento dos dados da mensagem.

**MQC.MQGMO\_ALL\_MSGS\_AVAILABLE\***

Recuperar mensagens de um grupo somente quando todas as mensagens no grupo estão disponíveis.

**MQC.MQGMO\_ALL\_SEGMENTS\_AVAILABLE\***

Recuperar os segmentos de uma mensagem lógica somente quando todos os segmentos no grupo estiverem disponíveis.

**MQC.MQGMO\_BROWSE\_FIRST**

Navegar a partir do início da fila.

**MQC.MQGMO\_BROWSE\_MSG\_UNDER\_CURSOR\***

Navegar pela mensagem sob o cursor de navegação.

**MQC.MQGMO\_BROWSE\_NEXT**

Navegar a partir da posição atual na fila.

**MQC.MQGMO\_COMPLETE\_MSG\***

Recuperar somente mensagens lógicas completas.

**MQC.MQGMO\_CONVERT**

Solicite que os dados do aplicativo sejam convertidos para que estejam em conformidade com os atributos CharSet e Encoding do MQMessage, antes que os dados sejam copiados para o buffer de mensagens Como a conversão de dados também é aplicada quando os dados são recuperados do buffer de mensagem, os aplicativos não configuram essa opção.

O uso dessa opção pode causar problemas ao converter de conjuntos de caracteres de byte único para conjuntos de caracteres de byte duplo Em vez disso, execute a conversão usando os métodos readString, readLinee writeString após a mensagem ser entregue.

**MQC.MQGMO\_FAIL\_IF QUIESCING**

Falha se o gerenciador de filas estiver inativo.

**MQC.MQGMO\_LOCK\***

Bloquear a mensagem que é navegada.

**MQC.MQGMO\_LOGICAL\_ORDER\***

Retornar mensagens em grupos e segmentos de mensagens lógicas, na ordem lógica.

Se você usar a opção MQGMO\_LOGICAL\_ORDER em um cliente reconectável, o código de razão MQRC\_RECONNECT\_INCOMPATIBLE será retornado para o aplicativo

**MQC.MQGMO\_MARK\_SKIP\_BACKOUT\***

Permitir que uma unidade de trabalho seja recuperada sem restabelecer a mensagem na fila.

**MQC.MQGMO\_MSG\_UNDER\_CURSOR**

Obter mensagem sob cursor de navegação.

**MQC.MQGMO\_NONE**

Nenhuma outra opção foi especificada; todas as opções assumem seus valores-padrão.

**MQC.MQGMO\_NO\_PROPERTIES**

Nenhuma propriedade da mensagem, exceto propriedades contidas no descritor de mensagens (ou extensão) é recuperada.

**MQC.MQGMO\_NO\_SYNCPOINT**

Obter mensagem sem controle de ponto de sincronização.

**MQC.MQGMO\_NO\_WAIT**

Retornar imediatamente se não houver mensagem adequada.

**MQC.MQGMO\_PROPERTIES\_AS\_Q\_DEF**

Recuperar propriedades de mensagem conforme definido pelo atributo `PropertyControl` de `MQQueue`. O acesso às propriedades de mensagem no descritor de mensagens ou extensão não é afetado pelo atributo `PropertyControl`.

**MQC.MQGMO\_PROPERTIES\_COMPATIBILITY**

Recuperar propriedades de mensagem com um prefixo de `mcd`, `jms`, `usj` ou `mjext`, nos cabeçalhos `MQRFH2`. Outras propriedades da mensagem, exceto as propriedades contidas no descritor de mensagem, ou extensão, são descartados.

**MQC.MQGMO\_PROPERTIES\_FORCE\_MQRFH2**

Recupere as propriedades de mensagem, exceto as propriedades contidas no descritor de mensagem ou extensão, nos cabeçalhos `MQRFH2`. Use `MQC.MQGMO_PROPERTIES_FORCE_MQRFH2` em aplicativos que estão esperando para recuperar propriedades, mas não podem ser alterados para usar identificadores de mensagem

**MQC.MQGMO\_PROPERTIES\_IN\_HANDLE**

Recuperar propriedades de mensagem usando um `MsgHandle`.

**MQC.MQGMO\_SYNCPOINT**

Obter a mensagem sob controle do ponto de sincronização. A mensagem é marcada como estando indisponível para outros aplicativos, mas é excluída da fila somente quando a unidade de trabalho é confirmada. A mensagem é disponibilizada novamente se a unidade de trabalho for recuperada.

**MQC.MQGMO\_SYNCPOINT\_IF\_PERSISTENT\***

Obter mensagem com controle de ponto de sincronização se a mensagem for persistente.

**MQC.MQGMO\_UNLOCK\***

Desbloquear uma mensagem bloqueada anteriormente.

**MQC.MQGMO\_WAIT**

Aguardar uma mensagem chegar.

**public string ResolvedQueueName {get;}**

O gerenciador de filas configura `ResolvedQueueName` para o nome local da fila da qual a mensagem foi recuperada. `ResolvedQueueName` é diferente do nome usado para abrir a fila se uma fila de alias ou fila modelo foi aberta.

**public char Segmentation {get;}\***

Segmentação indica se é possível permitir segmentação para a mensagem recuperada. Os valores possíveis são:

**MQC.MQSEG\_INHIBITED**

Não permitir a segmentação

**MQC.MQSEG\_ALLOWED**

Permitir segmentação

**public byte SegmentStatus {get;}\***

SegmentStatus é um campo de saída que indica se a mensagem recuperada é um segmento de uma mensagem lógica. Se a mensagem for um segmento, o sinalizador indicará se é o último segmento. Os valores possíveis são:

**MQC.MQSS\_LAST\_SEGMENT**

A mensagem é o último ou único segmento da mensagem lógica.

**MQC.MQSS\_NOT\_A\_SEGMENT**

A mensagem não é um segmento.

**MQC.MQSS\_SEGMENT**

A mensagem é um segmento, mas não é o último segmento da mensagem lógica.

**public int WaitInterval {get; set;}**

WaitInterval é o tempo máximo em milissegundos que uma chamada `MQQueue.get` aguarda uma mensagem adequada chegar. Use `WaitInterval` com `MQC.MQGMO_WAIT`. Configure um valor de `MQC.MQWI_UNLIMITED` para aguardar um tempo ilimitado para uma mensagem..

## Construtores

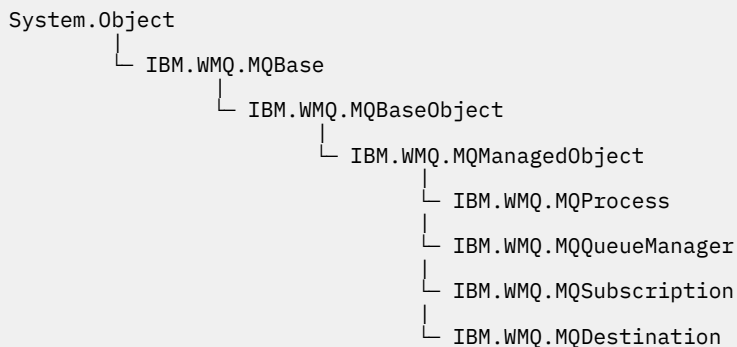
**public MQGetMessageOptions()**

Construa um novo objeto `MQGetMessageOptions` com `Options` configurado como `MQC.MQGMO_NO_WAIT`, `WaitInterval` configurado como zero e `ResolvedQueueName` configurado como em branco.

## Classe MQManagedObject.NET

Use `MQManagedObject` para consultar e definir os atributos de `MQDestination`, `MQProcess`, `MQQueueManager` e `MQSubscription`. `MQManagedObject` é uma superclasse dessas classes..

## Classes



```
public class IBM.WMQ.MQManagedObject extends IBM.WMQ.MQBaseObject;
```

- [“Propriedades” na página 1794](#)
- [“Métodos” na página 1795](#)
- [“Construtores” na página 1796](#)

## Propriedades

Teste para `MQException` ser lançado ao obter propriedades.

**public string AlternateUserId {get; set;}**

O ID do usuário alternativo, se houver, configurado quando o recurso foi aberto.

`AlternateUserID.set` é ignorado quando emitido para um objeto aberto. `AlternateUserId` não é válido para assinaturas.

**public int CloseOptions {get; set;}**

Configure esse atributo para controlar a maneira com que o recurso é fechado O valor padrão é MQC.MQCO\_NONE. MQC.MQCO\_NONE é o único valor permitido para todos os recursos diferentes de filas dinâmicas permanentes, filas dinâmicas temporárias, assinaturas e tópicos que estão sendo acessados pelos objetos que os criaram...

Para filas e tópicos, os valores adicionais a seguir são permitidos:

**MQC.MQCO\_DELETE**

Excluir a fila se não houver mensagens.

**MQC.MQCO\_DELETE\_PURGE**

Excluir a fila, limpando qualquer mensagem nela.

**MQC.MQCO QUIESCE**

Solicitar que a fila seja fechada, recebendo um aviso se quaisquer mensagens permanecerem (permitindo que elas sejam recuperadas antes do fechamento final).

Para assinaturas, os valores adicionais a seguir são permitidos:

**MQC.MQCO\_KEEP\_SUB**

A assinatura não é excluída. Esta opção é válida apenas se a assinatura original for durável. MQC.MQCO\_KEEP\_SUB é o valor padrão para um tópico durável.

**MQC.MQCO\_REMOVE\_SUB**

A assinatura será excluída. MQC.MQCO\_REMOVE\_SUB é o valor padrão para um tópico não durável e não gerenciado.

**MQC.MQCO\_PURGE\_SUB**

A assinatura será excluída. MQC.MQCO\_PURGE\_SUB é o valor padrão para um tópico gerenciado não durável.

**public MQQueueManager ConnectionReference {get;}**

O gerenciador de filas ao qual este recurso pertence.

**public string MQDescription {get;}**

A descrição do recurso conforme mantido pelo gerenciador de filas. MQDescription retorna uma sequência vazia para assinaturas e tópicos.

**public boolean IsOpen {get;}**

Indica se o recurso está aberto atualmente.

**public string Name {get;}**

O nome do recurso. O nome é fornecido no método de acesso ou o alocado pelo gerenciador de filas para uma fila dinâmica.

**public int OpenOptions {get; set;}**

OpenOptions são configurados quando um objeto IBM MQ é aberto O método OpenOptions.set é ignorado e não causa erro. Assinaturas não têm OpenOptions.

## Métodos

**public virtual void Close();**

Lança MQException..

Fecha o objeto. Nenhuma operação adicional com relação a esse recurso é permitida depois de chamar Close Para alterar o comportamento do método Close , configure o atributo closeOptions

**public string GetAttributeString(int selector, int length);**

Lança MQException..

Obtém uma cadeia de atributos.

**seletor**

Número inteiro indicando qual atributo está sendo consultado.

**comprimento**

Número inteiro indicando o comprimento da cadeia necessária.

**public void Inquire(int[] selectors, int[] intAttrs, byte[] charAttrs);**

Lança MQException..

Retorna uma matriz de números inteiros e um conjunto de sequências de caractere contendo os atributos de uma fila, processo ou gerenciador de filas. Os atributos a serem consultados são especificados na matriz de seletores.

**Nota:** Muitos dos atributos mais comuns podem ser consultados usando os métodos Get definidos em MQManagedObject, MQQueue e MQQueueManager..

**seletores**

Matriz de números inteiros que identifica os atributos com valores a serem consultados.

**intAttrs**

A matriz na qual os valores de atributos de número inteiro são retornados. Os valores de atributos de número inteiro são retornados na mesma ordem que os seletores de atributos de número inteiro na matriz de seletores.

**charAttrs**

O buffer no qual os atributos de caractere são retornados, concatenados. Os atributos de caracteres são retornados na mesma ordem que os seletores de atributos de caracteres na matriz de seletores. O comprimento de cada cadeia de atributos é fixada para cada atributo.

**public void Set(int[] selectors, int[] intAttrs, byte[] charAttrs);**

Lança MQException..

Configura os atributos definidos no vetor de seletores Os atributos a serem configurados são especificados na matriz de seletores.

**seletores**

Matriz de número inteiro que identifica os atributos com valores a serem configurados.

**intAttrs**

A matriz dos valores de atributos de número inteiro a serem configurados. Estes valores devem estar na mesma ordem que os seletores de atributos de número inteiro na matriz de seletores.

**charAttrs**

O buffer no qual os atributos de caractere a serem configurados são concatenados. Estes valores devem estar na mesma ordem que os seletores de atributos de caracteres na matriz de seletores. O comprimento de cada atributo de caractere é fixado.

**public void SetAttributeString(int selector, string value, int length);**

Lança MQException..

Configura uma cadeia de atributos.

**seletor**

Número inteiro indicando qual atributo está sendo configurado.

**Value**

A cadeia a configurar como o valor de atributo.

**comprimento**

Número inteiro indicando o comprimento da cadeia necessária.

## Construtores

**protected MQManagedObject()**

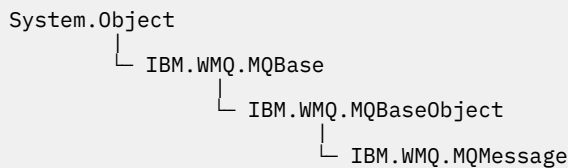
Método do construtor. Este objeto é uma classe base abstrata que não pode ser instanciada sozinha.

## Classe MQMessage.NET

Use MQMessage para acessar o descritor de mensagens e dados para uma mensagem IBM MQ . MQMessage contém uma mensagem IBM MQ .

### Parâmetro class





```
public class IBM.WMQ.MQMessage extends IBM.WMQ.MQBaseObject;
```

Crie um objeto `MQMessage` e, em seguida, use os métodos `Read` e `Write` para transferir dados entre a mensagem e outros objetos em seu aplicativo. Enviar e receber objetos do `MQMessage` usando os métodos `Put` e `Get` das classes `MQDestination`, `MQQueue` e `MQTopic`.

Obtenha e configure as propriedades do descritor de mensagens usando as propriedades de `MQMessage`. Configure e obtenha as propriedades de mensagem estendida usando os métodos `SetProperty` e `GetProperty`.

- [“Propriedades” na página 1797](#)
- [“Métodos de mensagem Read e Write” na página 1803](#)
- [“Métodos de buffer” na página 1806](#)
- [“Métodos de propriedade..” na página 1806](#)
- [“Construtores” na página 1809](#)

## Propriedades

Teste para `MQException` ser lançado ao obter propriedades.

```
public string AccountingToken {get; set;}
```

Parte do contexto de identidade da mensagem; ele ajuda um aplicativo a cobrar pelo trabalho feito como resultado da mensagem. O valor padrão é `MQC.MQACT_NONE`.

```
public string ApplicationIdData {get; set;}
```

Parte do contexto de identidade da mensagem.. `ApplicationId` são informações definidas pelo conjunto de aplicativos e podem ser usadas para fornecer informações adicionais sobre a mensagem ou seu originador. O valor padrão é "".

```
public string ApplicationOriginData {get; set;}
```

Informações definidas pelo aplicativo que podem ser usadas para fornecer informações adicionais sobre a origem da mensagem. O valor padrão é "".

```
public int BackoutCount {get;}
```

Uma contagem do número de vezes em que a mensagem foi retornada anteriormente e restaurada por uma chamada `MQQueue.Get` como parte de uma unidade de trabalho. O valor padrão é zero.

```
public int CharacterSet {get; set;}
```

O identificador do conjunto de caracteres codificados dos dados de caracteres na mensagem.

Configure `CharacterSet` para identificar o conjunto de caracteres de dados de caracteres na mensagem. Obtenha `CharacterSet` para descobrir em qual conjunto de caracteres foi usado para codificar os dados de caracteres na mensagem.

Os aplicativos .NET sempre são executados em Unicode, enquanto que em outros ambientes, os aplicativos são executados no mesmo conjunto de caracteres sob o qual o gerenciador de filas está sendo executado.

Os métodos `ReadString` e `ReadLine` convertem os dados de caractere na mensagem para Unicode.

O método `WriteString` converte de Unicode no conjunto de caracteres codificado em `CharacterSet`. Se `CharacterSet` for configurado para seu valor padrão, `MQC.MQCCSI_Q_MGR`, que é 0, nenhuma conversão ocorrerá e `CharacterSet` será configurado para 1200. Se você configurar `CharacterSet` para algum outro valor, `WriteString` converterá de Unicode para o valor alternativo.

**Nota:** Outros métodos de leitura e gravação não usam `CharacterSet`.

- ReadChar e WriteChar leem e gravam um caractere Unicode para e a partir do buffer de mensagem sem conversão.
- ReadUTF e WriteUTF convertem entre uma sequência Unicode no aplicativo e uma sequência UTF-8 , prefixada por um campo de comprimento de 2 bytes, no buffer de mensagem.
- Os bytes de transferência de métodos de byte entre o aplicativo e o buffer de mensagem sem alteração

**public byte[] CorrelationId {get; set;}**

- Para uma chamada MQQueue .Get , o identificador de correlação da mensagem a ser recuperada. O gerenciador de filas retorna a primeira mensagem com um identificador de mensagem e um identificador de correlação que correspondem aos campos do descritor de mensagem.. O valor padrão, MQC .MQCI\_NONE, ajuda qualquer identificador de correlação a corresponder.
- Para uma chamada MQQueue .Put , o identificador de correlação a ser configurado.

**public int DataLength {get;}**

O número de bytes de dados da mensagem restantes a serem lidos.

**public int DataOffset {get; set;}**

A posição atual do cursor dentro dos dados da mensagem Leituras e gravações entram em vigor na posição atual.

**public int Encoding {get; set;}**

A representação utilizada para valores numéricos nos dados da mensagem do aplicativo Codificação aplica-se a dados binários, decimais compactos e de ponto flutuante O comportamento dos métodos read e write para estes formatos numéricos é alterado de acordo. Construa um valor para o campo de codificação incluindo um valor de cada uma dessas três seções. Como alternativa, construa o valor combinando os valores de cada uma das três seções usando o operador OR bit a bit.

1. Inteiro Binário

**MQC.MQENC\_INTEGER\_NORMAL**

Números inteiros big endian.

**MQC.MQENC\_INTEGER\_REVERSED**

Inteiros little endian, conforme usados na arquitetura Intel .

2. Compactado-decimal

**MQC.MQENC\_DECIMAL\_NORMAL**

Decimal compactado big endian, conforme usado por z/OS.

**MQC.MQENC\_DECIMAL\_REVERSED**

Decimal compactado little endian.

3. ponto flutuante

**MQC.MQENC\_FLOAT\_IEEE\_NORMAL**

Flutuações IEEE big endian.

**MQC.MQENC\_FLOAT\_IEEE\_REVERSED**

O IEEE little endian flutua, conforme usado na arquitetura do Intel

**MQC.MQENC\_FLOAT\_S390**

z/OS formatação de pontos flutuantes

O valor padrão é:

```
MQC.MQENC_INTEGER_REVERSED |
MQC.MQENC_DECIMAL_REVERSED |
MQC.MQENC_FLOAT_IEEE_REVERSED
```

A configuração padrão faz com que o WriteInt grave um número inteiro little endian e o ReadInt leia um número inteiro little endian. Se você configurar a sinalização MQC .MQENC\_INTEGER\_NORMAL

sinalizador em vez disso, WriteInt grava um número inteiro big endian e ReadInt lê um número inteiro big endian.

**Nota:** Uma perda na precisão pode ocorrer ao converter pontos flutuantes do formato IEEE para pontos flutuantes do formato zSeries .

**public int Expiry {get; set;}**

Um tempo de expiração expresso em décimos de segundo, configurado pelo aplicativo que insere a mensagem. Depois que o tempo de expiração de uma mensagem tiver decorrido, ela será elegível para ser descartada pelo gerenciador de fila Se a mensagem tiver especificado um dos sinalizadores MQC.MQRO\_EXPIRATION , um relatório será gerado quando a mensagem for descartada O valor padrão é MQC.MQEI\_UNLIMITED, o que significa que a mensagem nunca expira.

**public int Feedback {get; set;}**

Use Feedback com uma mensagem do tipo MQC.MQMT\_REPORT para indicar a natureza do relatório.. Os códigos de feedback a seguir são definidos pelo sistema:

- MQC.MQFB\_EXPIRATION
- MQC.MQFB\_COA
- MQC.MQFB\_COD
- MQC.MQFB\_QUIT
- MQC.MQFB\_PAN
- MQC.MQFB\_NAN
- MQC.MQFB\_DATA\_LENGTH\_ZERO
- MQC.MQFB\_DATA\_LENGTH\_NEGATIVE
- MQC.MQFB\_DATA\_LENGTH\_TOO\_BIG
- MQC.MQFB\_BUFFER\_OVERFLOW
- MQC.MQFB\_LENGTH\_OFF\_BY\_ONE
- MQC.MQFB\_IIH\_ERROR

Os valores de feedback definidos pelo aplicativo no intervalo de MQC.MQFB\_APPL\_FIRST a MQC.MQFB\_APPL\_LAST também podem ser usados O valor padrão desse campo é MQC.MQFB\_NONE, indicando que nenhum feedback é fornecido.

**public string Format {get; set;}**

Um nome de formato usado pelo emissor da mensagem para indicar a natureza dos dados na mensagem para o receptor. É possível usar seus próprios nomes de formato, mas os nomes que começam com as letras MQ têm significados que são definidos pelo gerenciador de filas Os formatos integrados do gerenciador de filas são:

**MQC.MQFMT\_ADMIN**

Mensagem de pedido/resposta do servidor de comandos.

**MQC.MQFMT\_COMMAND\_1**

Mensagem de resposta do comando tipo 1.

**MQC.MQFMT\_COMMAND\_2**

Mensagem de resposta do comando tipo 2.

**MQC.MQFMT\_DEAD\_LETTER\_HEADER**

Cabeçalho da fila de devoluções.

**MQC.MQFMT\_EVENT**

Mensagem do evento.

**MQC.MQFMT\_NONE**

Nome sem formato.

**MQC.MQFMT\_PCF**

Mensagem definida pelo usuário no formato de comando programável.

**MQC.MQFMT\_STRING**

Mensagem consistindo totalmente em caracteres.

**MQC.MQFMT\_TRIGGER**

Mensagem do acionador

**MQC.MQFMT\_XMIT\_Q\_HEADER**

Cabeçalho da fila de transmissão.

O valor padrão é MQC.MQFMT\_NONE.

**public byte[] GroupId {get; set;}**

Uma cadeia de bytes que identifica o grupo de mensagens ao qual a mensagem física pertence. O valor padrão é MQC.MQGI\_NONE.

**public int MessageFlags {get; set;}**

Sinalizadores que controlam a segmentação e o status de uma mensagem.

**public byte[] MessageId {get; set;}**

Para uma chamada MQQueue.Get, esse campo especifica o identificador de mensagem da mensagem a ser recuperada. Normalmente, o gerenciador de filas retorna a primeira mensagem com um identificador de mensagem e identificador de correlação que correspondem aos campos do descritor de mensagem. Permita que qualquer identificador de mensagem corresponda usando o valor especial MQC.MQMI\_NONE.

Para uma chamada MQQueue.Put, esse campo especifica o identificador de mensagem a ser usado. Se MQC.MQMI\_NONE s especificado, o gerenciador de filas gera um identificador de mensagem exclusivo quando a mensagem é colocada. O valor desta variável é atualizada após a entrada, para indicar o identificador de mensagem que foi usado. O valor padrão é MQC.MQMI\_NONE.

**public int MessageLength {get;}**

O número de bytes de dados da mensagem no objeto MQMessage

**public int MessageSequenceNumber {get; set;}**

O número de sequência de uma mensagem lógica em um grupo.

**public int MessageType {get; set;}**

Indica o tipo da mensagem. Os valores a seguir são definidos atualmente pelo sistema:

- MQC.MQMT\_DATAGRAM
- MQC.MQMT\_REPLY
- MQC.MQMT\_REPORT
- MQC.MQMT\_REQUEST

Os valores definidos pelo aplicativo também podem ser usados, no intervalo de MQC.MQMT\_APPL\_FIRST a MQC.MQMT\_APPL\_LAST O valor-padrão desse campo é MQC.MQMT\_DATAGRAM.

**public int Offset {get; set;}**

Em uma mensagem segmentada, o deslocamento de dados em uma mensagem física a partir do início de uma mensagem lógica.

**public int OriginalLength {get; set;}**

O comprimento original de uma mensagem segmentada.

**public int Persistence {get; set;}**

Persistência da mensagem. Os seguintes valores são definidos:

- MQC.MQPER\_NOT\_PERSISTENT

Se você configurar essa opção em um cliente reconectável, o código de razão MQRC\_NONE será retornado para o aplicativo quando a conexão for bem-sucedida

- MQC.MQPER\_PERSISTENT

Se você configurar essa opção em um cliente reconectável, o código de razão MQRC\_CALL\_INTERRUPTED será retornado para o aplicativo após a conexão ser bem-sucedida

- MQC.MQPER\_PERSISTENCE\_AS\_Q\_DEF

O valor padrão é MQC.MQPER\_PERSISTENCE\_AS\_Q\_DEF, que obtém a persistência para a mensagem do atributo de persistência padrão da fila de destino.

**public int Priority {get; set;}**

A prioridade da mensagem. O valor especial MQC.MQPRI\_PRIORITY\_AS\_Q\_DEF também pode ser configurado na mensagem de saída. A prioridade para a mensagem é, então, obtida do atributo de prioridade padrão da fila de destino. O valor padrão é MQC.MQPRI\_PRIORITY\_AS\_Q\_DEF.

**public int PropertyValidation {get; set;}**

Especifica se a validação de propriedades ocorrerá quando uma propriedade da mensagem for configurada. Os valores possíveis são:

- MQCMHO\_DEFAULT\_VALIDATION
- MQCMHO\_VALIDATE
- MQCMHO\_NO\_VALIDATION

O valor padrão é MQCMHO\_DEFAULT\_VALIDATION.

**public string PutApplicationName {get; set;}**

O nome do aplicativo que coloca a mensagem. O valor padrão é "".

**public int PutApplicationType {get; set;}**

O tipo do aplicativo que gravou a mensagem. PutApplicationType pode ser um valor definido pelo sistema ou definido pelo usuário. Os valores a seguir são definidos pelo sistema:

- MQC.MQAT\_AIX
- MQC.MQAT\_CICS
- MQC.MQAT\_DOS
- MQC.MQAT\_IMS
- MQC.MQAT\_MVS
- MQC.MQAT\_OS2
- MQC.MQAT\_OS400
- MQC.MQAT\_QMGR
- MQC.MQAT\_UNIX
- MQC.MQAT\_WINDOWS
- MQC.MQAT\_JAVA

O valor padrão é MQC.MQAT\_NO\_CONTEXT, que indica que nenhuma informação de contexto está presente na mensagem.

**public DateTime PutDateTime {get; set;}**

A hora e data em que a mensagem foi colocada.

**public string ReplyToQueueManagerName {get; set;}**

O nome do gerenciador de filas para enviar mensagens de resposta ou de relatório. O valor padrão é "" e o gerenciador de filas fornece o ReplyToQueueManagerName.

**public string ReplyToQueueName {get; set;}**

O nome da fila de mensagem para a qual o aplicativo que emitiu a solicitação de obtenção para a mensagem envia mensagens MQC.MQMT\_REPLY e MQC.MQMT\_REPORT. O ReplyToQueueName padrão é "".

**public int Report {get; set;}**

Use Relatório para especificar opções sobre mensagens de relatório e resposta:

- Se relatórios são necessários.
- Se os dados da mensagem do aplicativo devem ser incluídos nos relatórios..
- Como configurar a mensagem e os identificadores de correlação no relatório ou resposta.

Qualquer combinação dos quatro tipos de relatórios pode ser solicitada:

- Especifique qualquer combinação dos quatro tipos de relatório Selecionar qualquer uma das três opções para cada tipo de relatório, dependendo se os dados da mensagem do aplicativo devem ser incluídos na mensagem de relatório.

1. Confirmar na chegada

- MQC.MQRO\_COA
- MQC.MQRO\_COA\_WITH\_DATA
- MQC.MQRO\_COA\_WITH\_FULL\_DATA \*\*

2. Confirmar na entrega

- MQC.MQRO\_COD
- MQC.MQRO\_COD\_WITH\_DATA
- MQC.MQRO\_COD\_WITH\_FULL\_DATA \*\*

3. Exceção

- MQC.MQRO\_EXCEPTION
- MQC.MQRO\_EXCEPTION\_WITH\_DATA
- MQC.MQRO\_EXCEPTION\_WITH\_FULL\_DATA \*\*

4. Expiração

- MQC.MQRO\_EXPIRATION
- MQC.MQRO\_EXPIRATION\_WITH\_DATA
- MQC.MQRO\_EXPIRATION\_WITH\_FULL\_DATA \*\*

**Nota:** Valores marcados com \*\* na lista não são suportados pelos gerenciadores de filas z/OS. Não use-os se for provável que seu aplicativo acesse um gerenciador de fila do z/OS, independentemente da plataforma na qual o aplicativo está em execução

- Especifique um dos seguintes para controlar como o ID de mensagem é gerado para a mensagem de relatório ou de resposta:
  - MQC.MQRO\_NEW\_MSG\_ID
  - MQC.MQRO\_PASS\_MSG\_ID
- Especifique um dos seguintes para controlar como o ID de correlação da mensagem de relatório ou de resposta deve ser configurado:
  - MQC.MQRO\_COPY\_MSG\_ID\_TO\_CORREL\_ID
  - MQC.MQRO\_PASS\_CORREL\_ID
- Especifique um dos seguintes para controlar a disposição da mensagem original quando ela não puder ser entregue à fila de destino:
  - MQC.MQRO\_DEAD\_LETTER\_Q
  - MQC.MQRO\_DISCARD\_MSG \*\*
- Se nenhuma opção de relatório for especificada, o padrão será:

```
MQC.MQRO_NEW_MSG_ID |
MQC.MQRO_COPY_MSG_ID_TO_CORREL_ID |
MQC.MQRO_DEAD_LETTER_Q
```

- É possível especificar um ou ambos do seguinte para solicitar que o aplicativo de recebimento envie uma mensagem de relatório de ação positiva ou de ação negativa.
  - MQC.MQRO\_PAN
  - MQC.MQRO\_NAN

**public int TotalMessageLength {get;}**

O número total de bytes na mensagem conforme armazenado na fila de mensagens a partir da qual esta mensagem foi recebida.

**public string UserId {get; set;}**

UserId faz parte do contexto de identidade da mensagem.. O gerenciador de filas geralmente fornece o valor. É possível substituir o valor se você tiver autoridade para configurar o contexto de identidade.

**public int Version {get; set;}**

A versão da estrutura MQMD em uso.

**Métodos de mensagem Read e Write**

Os métodos Read e Write executam as mesmas funções que os membros das classes BinaryReader e BinaryWriter no namespace .NET System. IO . Consulte o MSDN para obter a sintaxe de linguagem completa e exemplos de uso. Os métodos lidos ou gravados da posição atual no buffer de mensagem. Eles movem a posição atual para a frente pelo número de bytes lidos ou gravados

**Nota:** Se os dados da mensagem contiverem um cabeçalho MQRFH ou MQRFH2 , você deverá usar o método ReadBytes para ler os dados.

- Todos os métodos lançam IOException..
- Os métodos ReadFully redimensionam automaticamente a matriz de destino byte ou sbyte para ajustar a mensagem exatamente.. Uma matriz nula também é redimensionada.
- Read métodos throw EndOfStreamException.
- WriteDecimal métodos throw MQException.
- ReadString, ReadLine e WriteString métodos convertidos entre Unicode e o conjunto de caracteres da mensagem; consulte CharSet .
- Os métodos Decimal leem e gravam números decimais compactados codificados no formato big endian, MQC.MQENC\_DECIMAL\_NORMAL ou little endian MQC.MQENC\_DECIMAL\_REVERSE , de acordo com o valor de Encoding. Intervalos decimais e tipos .NET correspondentes são os seguintes:

**Decimal2/short**

-999 a 999

**Decimal4/int**

-9999999 a 9999999

**Decimal8/long**

-9999999999999999 a 9999999999999999

- Os métodos Double e Float leem e gravam valores flutuantes codificados nos formatos IEEE big endian e little endian, MQC.MQENC\_FLOAT\_IEEE\_NORMAL e MQC.MQENC\_FLOAT\_IEEE\_REVERSED ou no formato S/390 , MQC.MQENC\_FLOAT\_S390, de acordo com o valor de Encoding.
- Os métodos Int leem e gravam valores de número inteiro codificados no formato big endian, MQC.MQENC\_INTEGER\_NORMAL ou little endian, MQC.MQENC\_INTEGER\_REVERSED, de acordo com o valor de Encoding. Os números inteiros são todos assinados, exceto pela adição de um tipo de número inteiro de 2 bytes não assinado. Os tamanhos de número inteiro e os tipos .NET e IBM MQ são os seguintes:

**2 bytes**

short, Int2, ushort, UInt2

**4 bytes**

int, Int4

**8 bytes**

long, Int8

- WriteObject transfere a classe de um objeto, os valores de seus campos não temporários e não estáticos e os campos de seus supertipos para o buffer de mensagem.

- ReadObject cria um objeto da classe do objeto, a assinatura da classe e os valores de seus campos não temporários e não estáticos e os campos de seus supertipos.

<i>Tabela 843. Métodos de mensagem de leitura e gravação</i>	
<b>Tipo de destino</b>	<b>assinaturas do método</b>
<b>Boolean</b>	<pre>public bool ReadBoolean();  public void WriteBoolean(bool value);</pre>
<b>Byte</b>	<pre>public byte ReadByte() public byte ReadUnsignedByte()  public void Write(int value) public void WriteByte(int value) public void WriteByte(byte value) public void WriteByte(sbyte value)</pre>
<b>Bytes</b>	<pre>public byte[] ReadBytes(int count) public void ReadFully(ref byte[] value) public void ReadFully(ref sbyte[] value) public void ReadFully(ref byte[] value, int offset, int length) public void ReadFully(ref sbyte[] value, int offset, int length)  public void Write(byte[] value) public void Write(sbyte[] value) public void Write(byte[] value, int offset, int length) public void Write(sbyte[] value, int offset, int length) public void WriteBytes(string value)</pre>
<b>Decimal2</b>	<pre>public void WriteDecimal2(short value)</pre>
<b>Decimal4</b>	<pre>public void WriteDecimal4(short value)</pre>
<b>Decimal8</b>	<pre>public void WriteDecimal8(short value)</pre>
<b>Double</b>	<pre>public double ReadDouble()  public void WriteDouble(double value)</pre>
<b>Float</b>	<pre>public float ReadFloat()  public void WriteFloat(float value)</pre>



Tabela 843. Métodos de mensagem de leitura e gravação (continuação)

Tipo de destino	assinaturas do método
<b>Int2</b>	<pre>public void WriteInt2(int value)</pre>
<b>Int4</b>	<pre>public int readDecimal4() public int ReadInt() public int ReadInt4()  public void WriteInt(int value) public void WriteInt4(int value)</pre>
<b>Int8</b>	<pre>public void WriteInt8(long value)</pre>
<b>Long</b>	<pre>public long ReadDecimal8() public long ReadLong() public long ReadInt8()  public void WriteLong(long value)</pre>
<b>Object</b>	<pre>public Object ReadObject()  public void WriteObject(Object object)</pre>
<b>Short</b>	<pre>public short ReadShort() public short ReadDecimal2() public short ReadInt2()  public void WriteShort(int value)</pre>
<b>string</b>	<pre>public string ReadString(int length)  public void WriteString(string string)</pre>
<b>Unsigned Short</b>	<pre>public ushort ReadUnsignedShort() public ushort ReadUInt2()</pre>
<b>Unicode</b>	<pre>public string ReadLine() public char ReadChar()  public void WriteChar(int value) public void WriteChars(string string)</pre>

Tabela 843. Métodos de mensagem de leitura e gravação (continuação)

Tipo de destino	assinaturas do método
UTF	<pre>public string ReadUTF()  public void WriteUTF(string string)</pre>

## Métodos de buffer

### **public void ClearMessage();**

Lança `IOException`..

Descarta qualquer dado no buffer de mensagem e configura o deslocamento de dados de volta para zero.

### **public void ResizeBuffer(int size)**

Lança `IOException`..

Uma sugestão para o objeto `MQMessage` sobre o tamanho do buffer que pode ser necessário para operações de obtenção subsequentes. Se a mensagem contiver atualmente dados da mensagem e o novo tamanho for menor que o tamanho atual, os dados da mensagem serão truncados.

### **public void Seek(int pos)**

Lança `IOException`, `ArgumentOutOfRangeException`, `ArgumentException`.

Move o cursor para a posição absoluta no buffer de mensagem fornecido por *pos*. Leituras e gravações subsequentes atuam nesta posição no buffer.

### **public int SkipBytes(int i)**

Lança `IOException`, `EndOfStreamException`..

Avança *n* bytes no buffer de mensagem e retorna *n*, o número de bytes ignorados.

O método `SkipBytes` bloqueia até que ocorra um dos seguintes eventos:

- Todos os bytes são ignorados
- O fim do buffer de mensagem é detectado
- Uma exceção é lançada

## Métodos de propriedade..

### **public void DeleteProperty(string name);**

Lança `MQException`..

Exclui uma propriedade com o nome especificado da mensagem.

#### **nome**

O nome da propriedade a excluir.

### **public System.Collections.IEnumerator GetPropertyNames(string name)**

Lança `MQException`..

Retorna um `IEnumerator` de todos os nomes de propriedades que correspondem ao nome especificado. O sinal de porcentagem '%' pode ser usado no final do nome como um caractere curinga para filtrar as propriedades da mensagem, correspondendo a zero ou mais caracteres, incluindo o ponto.

**nome**

O nome da propriedade na qual corresponder.

**Métodos SetProperty e GetProperty**

Todos os métodos SetProperty e GetProperty lançam MQException

O método SetProperty da classe MQMessage .NET inclui uma nova propriedade se uma propriedade ainda não existir.. No entanto, se a propriedade já existir, o valor da propriedade fornecido será incluído no final da lista. Quando diversos valores são configurados para um nome de propriedade usando SetProperty, chamar GetProperty para esse nome retorna esses valores sequencialmente na ordem em que esses valores foram configurados

O comportamento é igual para todos os métodos Set\*Property e Get\*Property digitados, como GetLongProperty, SetLongProperty, GetBooleanProperty, SetBooleanProperty, GetStringProperty e SetStringProperty.

<i>Tabela 844. Métodos SetProperty e GetProperty</i>	
<b>tipo</b>	<b>assinaturas do método</b>
<b>Boolean</b>	<pre>public boolean GetBooleanProperty(string name); public boolean GetBooleanProperty(string name, MQPropertyDescriptor pd);  public void SetBooleanProperty(string name, boolean value); public void SetBooleanProperty(string name, MQPropertyDescriptor pd, boolean value);</pre>
<b>Byte</b>	<pre>public sbyte GetByteProperty(string name); public sbyte GetByteProperty(string name, MQPropertyDescriptor pd);  public void SetByteProperty(string name, sbyte value); public void SetByteProperty(string name, MQPropertyDescriptor pd, sbyte value);</pre>
<b>Bytes</b>	<pre>public sbyte[] GetBytesProperty(string name); public sbyte[] GetBytesProperty(string name, MQPropertyDescriptor pd);  public void SetBytesProperty(string name, sbyte[] value); public void SetBytesProperty(string name, MQPropertyDescriptor pd, sbyte[] value);</pre>
<b>Double</b>	<pre>public double GetDoubleProperty(string name); public double GetDoubleProperty(string name, MQPropertyDescriptor pd);  public void SetDoubleProperty(string name, double value); public void SetDoubleProperty(string name, MQPropertyDescriptor pd, double value);</pre>
<b>Float</b>	<pre>public float GetFloatProperty(string name); public float GetFloatProperty(string name, MQPropertyDescriptor pd);  public void SetFloatProperty(string name, float value); public void SetFloatProperty(string name, MQPropertyDescriptor pd, float value);</pre>

Tabela 844. Métodos SetProperty e GetProperty (continuação)

tipo	assinaturas do método
<b>Int2</b>	<pre>public short GetInt2Property(string name); public short GetInt2Property(string name, MQPropertyDescriptor pd);  public void SetInt2Property(string name, short value); public void SetInt2Property(string name, MQPropertyDescriptor pd, short value);</pre>
<b>Int4</b>	<pre>public int GetInt4Property(string name); public int GetInt4Property(string name, MQPropertyDescriptor pd);  public void SetInt4Property(string name, int value); public void SetInt4Property(string name, MQPropertyDescriptor pd, int value);</pre>
<b>Int8</b>	<pre>public long GetInt8Property(string name); public long GetInt8Property(string name, MQPropertyDescriptor pd);  public void SetInt8Property(string name, long value); public void SetInt8Property(string name, MQPropertyDescriptor pd, long value);</pre>
<b>Long</b>	<pre>public long GetLongProperty(string name); public long GetLongProperty(string name, MQPropertyDescriptor pd);  public void SetLongProperty(string name, long value); public void SetLongProperty(string name, MQPropertyDescriptor pd, long value);</pre>
<b>Object</b>	<pre>public Object GetObjectProperty(string name); public Object GetObjectProperty(string name, MQPropertyDescriptor pd);  public void SetObjectProperty(string name, Object value); public void SetObjectProperty(string name, MQPropertyDescriptor pd, Object value);</pre>
<b>Short</b>	<pre>public short GetShortProperty(string name); public short GetShortProperty(string name, MQPropertyDescriptor pd);  public void SetShortProperty(string name, short value); public void SetShortProperty(string name, MQPropertyDescriptor pd, short value);</pre>
<b>string</b>	<pre>public string GetStringProperty(string name); public string GetStringProperty(string name, MQPropertyDescriptor pd);  public void SetStringProperty(string name, string value); public void SetStringProperty(string name, MQPropertyDescriptor pd, string value);</pre>

## Construtores

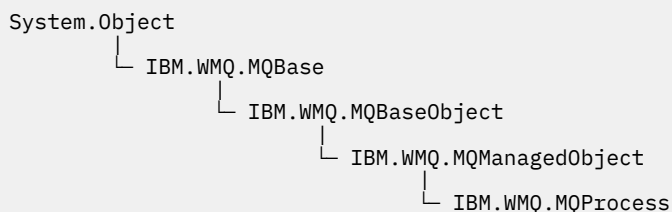
**public MQMessage();**

Cria um objeto MQMessage com as informações do descritor de mensagens padrão e um buffer de mensagem vazio

## Classe MQProcess.NET

Use MQProcess para consultar os atributos de um processo do IBM MQ Crie um objeto MQProcess usando um construtor, ou um método MQQueueManager AccessProcess

### Parâmetro class



```
public class IBM.WMQ.MQProcess extends IBM.WMQ.MQManagedObject;
```

- [“Propriedades” na página 1809](#)
- [“Construtores” na página 1810](#)

### Propriedades

Teste para MQException ser lançado ao obter propriedades.

**public string ApplicationId {get;}**

Obtém a cadeia de caracteres que identifica o aplicativo a ser iniciado. ApplicationId é usado por um aplicativo monitor acionador. ApplicationId é enviado à fila de inicialização como parte da mensagem do acionador.

O valor padrão é nulo.

**public int ApplicationType {get;}**

Identifica o tipo do processo a ser iniciado por um aplicativo de monitor acionador Tipos padrão são definidos, mas outros podem ser usados:

- MQAT\_AIX
- MQAT\_CICS
- MQAT\_IMS
- MQAT\_MVS
- MQAT\_NATIVE
- MQAT\_OS400
- MQAT\_UNIX
- MQAT\_WINDOWS
- MQAT\_JAVA
- MQAT\_USER\_FIRST
- MQAT\_USER\_LAST

O valor padrão é MQAT\_NATIVE.

```
public string EnvironmentData {get;}
```

Obtém informações sobre o ambiente do aplicativo que será iniciado.

O valor padrão é nulo.

```
public string UserData {get;}
```

Obtém informações fornecidas pelo usuário sobre o aplicativo a ser iniciado.

O valor padrão é nulo.

## Construtores

```
public MQProcess(MQQueueManager queueManager, string processName, int openOptions);
```

```
public MQProcess(MQQueueManager qMgr, string processName, int openOptions, string queueManagerName, string alternateUserId);
```

Lança MQException..

Acesse um processo IBM MQ no gerenciador de fila *qMgr* para consultar atributos de processo.

### **qMgr**

Gerenciador de filas para acessar.

### **processName**

O nome do processo a abrir.

### **openOptions**

Opções que controlam a abertura do processo. As opções válidas que podem ser incluídas ou combinadas usando um OR bit a bit são:

- MQC.MQ00\_FAIL\_IF QUIESCING
- MQC.MQ00\_INQUIRE
- MQC.MQ00\_SET
- MQC.MQ00\_ALTERNATE\_USER\_AUTHORITY

### **queueManagerName**

O nome do gerenciador de filas no qual o processo está definido. É possível deixar um nome de gerenciador de fila em branco ou nulo se o gerenciador de filas for igual ao que o processo está acessando.

### **alternateUserId**

Se MQC.MQ00\_ALTERNATE\_USER\_AUTHORITY for especificado no parâmetro **openOptions**, *alternateUserId* especificará o ID do usuário alternativo usado para verificar a autorização para a ação. Se MQ00\_ALTERNATE\_USER\_AUTHORITY não for especificado, *alternateUserId* poderá ficar em branco ou nulo.

A autoridade do usuário padrão será usada para a conexão com o gerenciador de fila se MQC.MQ00\_ALTERNATE\_USER\_AUTHORITY não for especificado.

```
public MQProcess MQQueueManager.AccessProcess(string processName, int openOptions);
```

```
public MQProcess MQQueueManager.AccessProcess(string processName, int openOptions, string queueManagerName, string alternateUserId);
```

Lança MQException..

Acesse um processo IBM MQ neste gerenciador de filas para consultar atributos de processo.

### **processName**

O nome do processo a abrir.

### openOptions

Opções que controlam a abertura do processo. As opções válidas que podem ser incluídas ou combinadas usando um OR bit a bit são:

- MQC.MQOO\_FAIL\_IF QUIESCING
- MQC.MQOO\_INQUIRE
- MQC.MQOO\_SET
- MQC.MQOO\_ALTERNATE\_USER\_AUTHORITY

### queueManagerName

O nome do gerenciador de filas no qual o processo está definido. É possível deixar um nome de gerenciador de fila em branco ou nulo se o gerenciador de filas for igual ao que o processo está acessando.

### alternateUserId

Se MQC.MQOO\_ALTERNATE\_USER\_AUTHORITY for especificado no parâmetro **openOptions**, *alternateUserId* especificará o ID do usuário alternativo usado para verificar a autorização para a ação. Se MQOO\_ALTERNATE\_USER\_AUTHORITY não for especificado, *alternateUserId* poderá ficar em branco ou nulo.

A autoridade do usuário padrão será usada para a conexão com o gerenciador de fila se MQC.MQOO\_ALTERNATE\_USER\_AUTHORITY não for especificado.

## Classe MQPropertyDescriptor.NET

Use MQPropertyDescriptor como um parâmetro para MQMessage.GetProperty e SetProperty métodos. MQPropertyDescriptor descreve uma propriedade MQMessage ..

### Parâmetro class

```
System.Object
└─ IBM.WMQ.MQPropertyDescriptor
```

```
public class IBM.WMQ.MQPropertyDescriptor extends System.Object;
```

- [“Propriedades” na página 1811](#)
- [“Construtores” na página 1812](#)

### Propriedades

Teste para MQException ser lançado ao obter propriedades.

```
public int Context {get; set;}
```

O contexto da mensagem ao qual a propriedade pertence. Os valores possíveis são:

#### MQC.MQPD\_NO\_CONTEXT

A propriedade não é associada com um contexto de mensagem.

#### MQC.MQPD\_USER\_CONTEXT

A propriedade é associada com o contexto do usuário.

Se o usuário estiver autorizado, uma propriedade associada ao contexto do usuário será salva quando uma mensagem for recuperada.. Um método Put subsequente referenciando o contexto salvo pode passar a propriedade para a nova mensagem

```
public int CopyOptions {get; set;}
```

CopyOptions descreve em qual tipo de mensagem a propriedade pode ser copiada para.

Quando um gerenciador de filas recebe uma mensagem contendo uma propriedade definida IBM MQ que o gerenciador de filas reconhece como incorreta, o gerenciador de filas corrige o valor do campo CopyOptions.

Qualquer combinação das opções a seguir pode ser especificada Combine as opções incluindo os valores ou usando o bit a bit OR

#### **MQC.MQCOPY\_ALL**

A propriedade é copiada em todas as mensagens subsequentes.

#### **MQC.MQCOPY\_FORWARD**

A propriedade é copiada em uma mensagem sendo encaminhada.

#### **MQC.MQCOPY\_PUBLISH**

A propriedade é copiada para a mensagem recebida por um assinante quando uma mensagem está sendo publicada

#### **MQC.MQCOPY\_REPLY**

A propriedade é copiada em uma mensagem de resposta.

#### **MQC.MQCOPY\_REPORT**

A propriedade é copiada em uma mensagem de relatório

#### **MQC.MQCOPY\_DEFAULT**

O valor indicado não foram especificadas outras opções de cópia. Não existe relacionamento entre a propriedade e as mensagens subsequentes. MQC.MQCOPY\_DEFAULT é sempre retornado para propriedades do descritor de mensagens.

#### **MQC.MQCOPY\_NONE**

O mesmo que MQC.MQCOPY\_DEFAULT

```
public int Options { set; }
```

Opções é padronizado para CMQC.MQPD\_NONE Não é possível configurar nenhum outro valor.

```
public int Support { get; set; }
```

Configure Support para especificar o nível de suporte necessário para propriedades de mensagem definidas pelo IBM MQ. O suporte para todas as outras propriedades é opcional Qualquer ou nenhum dos valores a seguir pode ser especificado

#### **MQC.MQPD\_SUPPORT\_OPTIONAL**

A propriedade é aceita por um gerenciador de filas mesmo se ela não for suportada. A propriedade pode ser descartada para a mensagem fluir para um gerenciador de filas que não suporta propriedades de mensagens. Esse valor também é designado a propriedades que não são IBM MQ definidas.

#### **MQC.MQPD\_SUPPORT\_REQUIRED**

É necessário suporte para a propriedade. Se você colocar a mensagem em um gerenciador de fila que não suporta a propriedade definida pelo IBM MQ, o método falhará Ele retorna o código de conclusão MQC.MQCC\_FAILED e o código de razão MQC.MQRC\_UNSUPPORTED\_PROPERTY

#### **MQC.MQPD\_SUPPORT\_REQUIRED\_IF\_LOCAL**

O suporte para a propriedade será necessário, se a mensagem for destinada para uma fila local Se você colocar a mensagem em uma fila local em um gerenciador de filas que não suporta a propriedade definida pelo IBM MQ, o método falhará Ele retorna o código de conclusão MQC.MQCC\_FAILED e o código de razão MQC.MQRC\_UNSUPPORTED\_PROPERTY

Não será feita nenhuma verificação se a mensagem for colocada em um gerenciador de filas remotas

## **Construtores**

```
PropertyDescriptor();
```

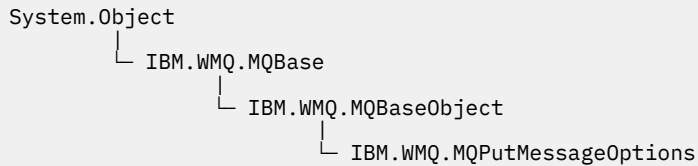
Crie um descritor de propriedade



## Classe MQPutMessageOptions.NET

Use MQPutMessageOptions para especificar como as mensagens são enviadas Ele modifica o comportamento do MQDestination.Put

### Parâmetro class



```
public class IBM.WMQ.MQPutMessageOptions extends IBM.WMQ.MQBaseObject;
```

- [“Propriedades” na página 1813](#) [“Construtores” na página 1815](#)

### Propriedades

Teste para MQException ser lançado ao obter propriedades.

**Nota:** O comportamento de algumas das opções disponíveis nesta classe depende do ambiente no qual elas são usadas. Estes elementos são marcados com um asterisco, \*.

#### **public MQQueue ContextReference {get; set;}**

Se o campo options incluir MQC.MQPMO\_PASS\_IDENTITY\_CONTEXT ou MQC.MQPMO\_PASS\_ALL\_CONTEXT, configure esse campo para se referir ao MQQueue do qual obter as informações de contexto.

O valor inicial deste campo é nulo.

#### **public int InvalidDestCount {get;} \***

Geralmente, usado para listas de distribuição, InvalidDestCount indica o número de mensagens que não puderam ser enviadas para as filas em uma lista de distribuição A contagem inclui filas que falharam ao abrir e também as filas que foram abertas com êxito, mas para as quais a operação de colocação falhou

.NET não suporta listas de distribuição, mas InvalidDestCount é configurado ao abrir uma única fila.

#### **public int KnownDestCount {get;} \***

Geralmente usado para listas de distribuição, KnownDestCount indica o número de mensagens que a chamada atual enviou com êxito para filas que são resolvidas para filas locais.

.NET não suporta listas de distribuição, mas InvalidDestCount é configurado ao abrir uma única fila.

#### **public int Options {get; set;}**

Opções que controlam a ação de MQDestination.put e MQQueueManager.put. Qualquer ou nenhum dos valores a seguir pode ser especificado. Se mais de uma opção for necessária, os valores poderão ser incluídos ou combinados usando o operador OR bit a bit..

##### **MQC.MQPMO\_ASYNC\_RESPONSE**

Essa opção faz com que a chamada MQDestination.put seja feita assincronamente, com alguns dados de resposta.

##### **MQC.MQPMO\_DEFAULT\_CONTEXT**

Associar contexto padrão à mensagem.

**MQC.MQPMO\_FAIL\_IF QUIESCING**

Falha se o gerenciador de filas estiver inativo.

**MQC.MQPMO\_LOGICAL\_ORDER \***

Colocar mensagens lógicas e segmentos nos grupos de mensagens em suas ordens lógicas.

Se você usar a opção MQPMO\_LOGICAL\_ORDER em um cliente reconectável, o código de razão MQRC\_RECONNECT\_INCOMPATIBLE será retornado para o aplicativo

**MQC.MQPMO\_NEW\_CORREL\_ID \***

Gerar um novo ID de correlação para cada mensagem enviada.

**MQC.MQPMO\_NEW\_MSG\_ID \***

Gerar um novo ID de mensagem para cada mensagem enviada.

**MQC.MQPMO\_NONE**

Nenhuma opção especificada. Não use com outras opções..

**MQC.MQPMO\_NO\_CONTEXT**

Nenhum contexto deve ser associado à mensagem.

**MQC.MQPMO\_NO\_SYNCPOINT**

Coloque uma mensagem sem o controle do ponto de sincronização Se a opção de controle de ponto de sincronização não for especificada, um padrão de nenhum ponto de sincronização será assumido

**MQC.MQPMO\_PASS\_ALL\_CONTEXT**

Transmitir todo contexto a partir de um identificador de filas de entrada.

**MQC.MQPMO\_PASS\_IDENTITY\_CONTEXT**

Transmitir contexto de identidade a partir de um identificador de filas de entrada.

**MQC.MQPMO\_RESPONSE\_AS\_Q\_DEF**

Para uma chamada MQDestination.put, esta opção usa o tipo de resposta put do atributo DEFPRESP da fila

Para uma chamada MQQueueManager.put, essa opção faz a chamada ser feita de forma síncrona.

**MQC.MQPMO\_RESPONSE\_AS\_TOPIC\_DEF**

MQC.MQPMO\_RESPONSE\_AS\_TOPIC\_DEF é um sinônimo para MQC.MQPMO\_RESPONSE\_AS\_Q\_DEF para uso com objetos de tópico.

**MQC.MQPMO\_RETAIN**

A publicação que está sendo enviada deve ser retida pelo gerenciador de filas. Se essa opção for usada e a publicação não puder ser retida, a mensagem não será publicada e a chamada falhará com MQC.MQRC\_PUT\_NOT\_RETAINED

Solicite uma cópia desta publicação após o horário em que ela foi publicada, chamando o método MQSubscription.RequestPublicationUpdate A publicação salva é enviada para os aplicativos que criam uma assinatura sem configurar a opção MQC.MQSO\_NEW\_PUBLICATIONS\_ONLY Verifique a propriedade de mensagem MQIsRetained de uma publicação, quando ela for recebida, para descobrir se ela foi a publicação retida

Quando as publicações retidas são solicitadas por um assinante, a assinatura usada pode conter um curinga na sequência de tópicos Se houver várias publicações retidas na árvore de tópicos que correspondem à assinatura, todas elas serão enviadas.

**MQC.MQPMO\_SET\_ALL\_CONTEXT**

Configurar todo contexto a partir do aplicativo.

**MQC.MQPMO\_SET\_IDENTITY\_CONTEXT**

Configurar contexto de identidade a partir do aplicativo.

**MQC.MQPMO\_SYNC\_RESPONSE**

Essa opção faz com que a chamada MQDestination.put ou MQQueueManager.put seja feita de forma síncrona, com dados de resposta completos..

### **MQC.MQPMO\_SUPPRESS\_REPLYTO**

Quaisquer informações preenchidas nos campos ReplyToQueueName e ReplyToQueueManagerName da publicação não são transmitidas aos assinantes. Se essa opção for usada em combinação com uma opção de relatório que requer um ReplyToQueueName, a chamada falhará com MQC.MQRC\_MISSING\_REPLY\_TO\_Q.

### **MQC.MQPMO\_SYNCPOINT**

Coloque uma mensagem com o controle de ponto de sincronização A mensagem não é visível fora da unidade de trabalho até que a unidade de trabalho seja confirmada. Se a unidade de trabalho for recuperada, a mensagem será excluída.

### **public int RecordFields {get; set;} \***

Informações sobre listas de distribuição. As listas de distribuição não estão suportando no .NET

### **public string ResolvedQueueManagerName {get;}**

Um campo de saída configurado pelo gerenciador de filas com o nome do gerenciador de filas que possui a fila especificada pelo nome da fila remota. ResolvedQueueManagerName pode ser diferente do nome do gerenciador de filas do qual a fila foi acessada se a fila for uma fila remota.

Um valor não em branco será retornado apenas se o objeto for uma única fila.. Se o objeto for uma lista de distribuição ou um tópico, o valor retornado será indefinido.

### **public string ResolvedQueueName {get;}**

Um campo de saída que é configurado pelo gerenciador de filas com o nome da fila na qual a mensagem é colocada. ResolvedQueueName pode ser diferente do nome usado para abrir a fila se a fila aberta era um alias ou fila modelo.

Um valor não em branco será retornado apenas se o objeto for uma única fila.. Se o objeto for uma lista de distribuição ou um tópico, o valor retornado será indefinido.

### **public int UnknownDestCount {get;} \***

Geralmente usado para listas de distribuição, UnknownDestCount é um campo de saída configurado para o gerenciador de filas Ele relata o número de mensagens que a chamada atual enviou com êxito para as filas que são resolvidas para filas remotas

.NET não suporta listas de distribuição, mas InvalidDestCount é configurado ao abrir uma única fila.

## **Construtores**

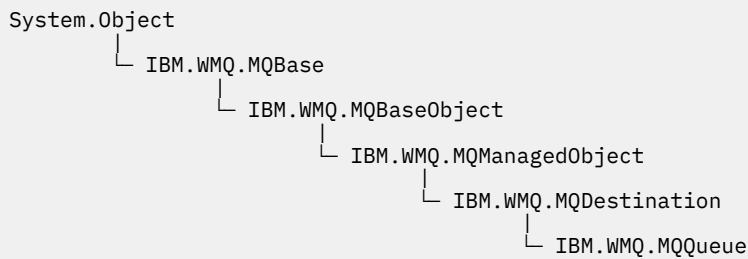
### **public MQPutMessageOptions();**

Construa um novo objeto MQPutMessageOptions sem opções configuradas e um ResolvedQueueName e ResolvedQueueManagerName em branco.

## **Classe MQQueue.NET**

Use MQQueue para enviar e receber mensagens e consultar atributos de uma fila IBM MQ . Crie um objeto MQQueue usando um construtor, ou um método MQQueueManager . AccessProcess

## **Parâmetro class**



```
public class IBM.WMQ.MQQueue extends IBM.WMQ.MQDestination;
```

- [“Propriedades” na página 1816](#)
- [“Métodos” na página 1818](#)
- [“Construtores” na página 1820](#)

## Propriedades

Teste para MQException ser lançado ao obter propriedades.

```
public int ClusterWorkLoadPriority {get;}
```

Especifica a prioridade da fila. Este parâmetro é válido somente para filas locais, remotas e de alias.

```
public int ClusterWorkLoadRank {get;}
```

Especifica a classificação da fila. Este parâmetro é válido somente para filas locais, remotas e de alias.

```
public int ClusterWorkLoadUseQ {get;}
```

Especifica o comportamento de uma operação MQPUT quando a fila de destino possui uma instância local e pelo menos uma instância de cluster remota. Este parâmetro não se aplica se MQPUT se originar de um canal de cluster. Este parâmetro é válido somente para filas locais.

```
public DateTime CreationDateTime {get;}
```

A data e hora em que esta fila foi criada.

```
public int CurrentDepth {get;}
```

Obtém o número de mensagens atualmente na fila. Este valor é incrementado durante uma chamada put e durante a restauração de uma chamada get. Ele é decrementado durante uma chamada get de não navegação e durante a restauração de uma chamada put.

```
public int DefinitionType {get;}
```

Como a fila foi definida. Os valores possíveis são:

- MQC.MQQDT\_PREDEFINED
- MQC.MQQDT\_PERMANENT\_DYNAMIC
- MQC.MQQDT\_TEMPORARY\_DYNAMIC

```
public int InhibitGet {get; set;}
```

Controla se é possível obter mensagens nesta fila ou para este tópico. Os valores possíveis são:

- MQC.MQQA\_GET\_INHIBITED
- MQC.MQQA\_GET\_ALLOWED

```
public int InhibitPut {get; set;}
```

Controla se você pode colocar mensagens nesta fila ou para este tópico. Os valores possíveis são:

- MQQA\_PUT\_INHIBITED
- MQQA\_PUT\_ALLOWED

```
public int MaximumDepth {get;}
```

O número máximo de mensagens que podem existir na fila a qualquer momento. Uma tentativa de colocar uma mensagem em uma fila que já contém essas mensagens falha com o código de razão MQC.MQRC\_Q\_FULL.

**public int MaximumMessageLength {get;}**

O comprimento máximo dos dados do aplicativo que podem existir em cada mensagem nesta fila. Uma tentativa de colocar uma mensagem maior que esse valor falha com o código de razão MQC.MQRC\_MSG\_TOO\_BIG\_FOR\_Q..

**public int NonPersistentMessageClass {get;}**

O nível de confiabilidade para mensagens não persistentes colocadas nesta fila.

**public int OpenInputCount {get;}**

O número de identificadores que são atualmente válidos para remover mensagens da fila. OpenInputCount é o número total de identificadores de entrada válidos conhecidos para o gerenciador de filas locais, não apenas identificadores criados pelo aplicativo.

**public int OpenOutputCount {get;}**

O número de identificadores que são atualmente válidos para incluir mensagens na fila. OpenOutputCount é o número total de identificadores de saída válidos conhecidos para o gerenciador de filas locais, não apenas identificadores criados pelo aplicativo.

**public int QueueAccounting {get;}**

Especifica se é possível ativar a coleção de informações de contabilidade para a fila.

**public int QueueMonitoring {get;}**

Especifica se é possível ativar o monitoramento para a fila

**public int QueueStatistics {get;}**

Especifica se você pode ativar a coleta de estatísticas para a fila

**public int QueueType {get;}**

O tipo desta fila com um dos seguintes valores:

- MQC.MQQT\_ALIAS
- MQC.MQQT\_LOCAL
- MQC.MQQT\_REMOTE
- MQC.MQQT\_CLUSTER

**public int Shareability {get;}**

Se a fila pode ser aberta para entrada múltiplas vezes. Os valores possíveis são:

- MQC.MQQA\_SHAREABLE
- MQC.MQQA\_NOT\_SHAREABLE

**public string TPIPE {get;}**

O nome TPIPE usado para comunicação com o OTMA usando a Ponte do IBM MQ IMS

**public int TriggerControl {get; set;}**

Se as mensagens do acionador são gravadas em uma fila de inicialização, para iniciar um aplicativo para servir a fila. Os valores possíveis são:

- MQC.MQTC\_OFF
- MQC.MQTC\_ON

**public string TriggerData {get; set;}**

Os dados de formato livre que o gerenciador de filas insere na mensagem acionador. Ele insere TriggerData quando uma mensagem que chega nessa fila faz com que uma mensagem do acionador seja gravada na fila de iniciação O comprimento máximo permitido da sequência é fornecido por MQC.MQ\_TRIGGER\_DATA\_LENGTH

**public int TriggerDepth {get; set;}**

O número de mensagens que devem estar na fila antes que uma mensagem do acionador seja gravada quando o tipo de acionador for configurado como MQC.MQTT\_DEPTH

**public int TriggerMessagePriority {get; set;}**

A prioridade da mensagem sob a qual as mensagens não contribuem para a geração de mensagens do acionador Ou seja, o gerenciador de filas ignora essas mensagens ao decidir se um acionador deve ser gerado. Um valor igual a zero faz com que todas as mensagens contribuam com a geração de mensagens do acionador.

```
public int TriggerType {get; set;}
```

As condições sob as quais mensagens do acionador são gravadas como resultado de mensagens que chegam nesta fila. Os valores possíveis são:

- MQC.MQTT\_NONE
- MQC.MQTT\_FIRST
- MQC.MQTT\_EVERY
- MQC.MQTT\_DEPTH

## Métodos

```
public void Get(MQMessage message);
```

```
public void Get(MQMessage message, MQGetMessageOptions getMessageOptions);
```

```
public void Get(MQMessage message, MQGetMessageOptions getMessageOptions, int MaxMsgSize);
```

Lança MQException..

Obtém uma mensagem de uma fila.

Se a obtenção falhar, o objeto MQMessage permanecerá inalterado. Se ele for bem-sucedido, as partes do descritor de mensagens e de dados da mensagem do MQMessage serão substituídas pelo descritor de mensagens e pelos dados da mensagem recebida.

Todas as chamadas para IBM MQ de um determinado MQQueueManager são síncronas. Portanto, se você executar um get com wait, todos os outros encadeamentos usando o mesmo MQQueueManager serão bloqueados de fazer chamadas IBM MQ adicionais até que a chamada Get seja realizada.. Se você precisar de vários encadeamentos para acessar o IBM MQ simultaneamente, cada encadeamento deverá criar seu próprio objeto MQQueueManager

### mensagem

Contém o descritor de mensagens e os dados da mensagem retornados. Alguns dos campos no descritor de mensagem são parâmetros de entrada. É importante assegurar que os parâmetros de entrada MessageId e CorrelationId sejam configurados conforme necessário.

Um cliente reconectável retorna o código de razão MQRC\_BACKED\_OUT após a reconexão bem-sucedida para mensagens recebidas em MQGM\_SYNCPOINT.

### Opções getMessage

Opções que controlam a ação de obtenção.

Usar a opção MQC.MQGM\_CONVERT pode resultar em uma exceção com o código de razão MQC.MQRC\_CONVERTED\_STRING\_TOO\_BIG ao converter de códigos de caracteres de byte único para códigos de byte duplo. Neste caso, a mensagem é copiada para o buffer sem conversão.

Se *getMessageOptions* não for especificado, a opção de mensagem usada será MQGM\_NOWAIT.

Se você usar a opção MQGM\_LOGICAL\_ORDER em um cliente reconectável, o código de razão MQRC\_RECONNECT\_INCOMPATIBLE será retornado.

### MaxMsgTamanho

A maior mensagem que este objeto de mensagem deve receber. Se a mensagem na fila for maior que este tamanho, uma de duas coisas ocorrerá:

- Se a sinalização MQGM\_ACCEPT\_TRUNCATED\_MSG for configurada no objeto MQGetMessageOptions, a mensagem será preenchida com o máximo de dados da mensagem possível. Uma exceção é lançada com o código de conclusão MQCC\_WARNING e o código de razão MQRC\_TRUNCATED\_MSG\_ACCEPTED.
- Se o sinalizador MQGM\_ACCEPT\_TRUNCATED\_MSG não for configurado, a mensagem permanecerá na fila. Uma exceção é lançada com o código de conclusão MQCC\_WARNING e o código de razão MQRC\_TRUNCATED\_MSG\_FAILED.

Se *MaxMsgSize* não for especificado, a mensagem inteira será recuperada.

```
public void Put(MQMessage message);  
public void Put(MQMessage message, MQPutMessageOptions putMessageOptions);
```

Lança `MQException`.

Coloca uma mensagem em uma fila.

Modificações no objeto `MQMessage` após a chamada `Put` ter sido realizada não afetam a mensagem real na fila IBM MQ ou tópico de publicação.

`Put` atualiza as propriedades `MessageId` e `CorrelationId` do objeto `MQMessage` e não limpa dados da mensagem. Chamadas `Put` ou `Get` adicionais se referem às informações atualizadas no objeto `MQMessage`. Por exemplo, no fragmento de código a seguir, a primeira mensagem contém a e a segunda ab.

```
msg.WriteString("a");  
q.Put(msg, pmo);  
msg.WriteString("b");  
q.Put(msg, pmo);
```

### **mensagem**

Um objeto `MQMessage` que contém os dados do descritor de mensagens e a mensagem a ser enviada. O descritor de mensagens pode ser alterado como uma consequência deste método. Os valores no descritor de mensagens imediatamente após a conclusão desse método são os valores que foram colocados na fila ou publicados no tópico.

Os seguintes códigos de razão são retornados para um cliente reconectável:

- `MQRC_CALL_INTERRUPTED` se a conexão for interrompida ao executar uma chamada `Put` em uma mensagem persistente e a reconexão for bem-sucedida.
- `MQRC_NONE` se a conexão for bem-sucedida ao executar uma chamada `Put` em uma mensagem não persistente (consulte [Recuperação de Aplicativo](#)).

### **putMessageOptions**

Opções que controlam a ação da entrada.

Se `putMessageOptions` não for especificado, a instância padrão de `MQPutMessageOptions` será usada.

Se você usar a opção `MQPMO_LOGICAL_ORDER` em um cliente reconectável, o código de razão `MQRC_RECONNECT_INCOMPATIBLE` será retornado.

**Nota:** Para simplicidade e desempenho, se você desejar colocar uma única mensagem em uma fila, use o objeto `MQQueueManager.Put`. Deve-se ter um objeto `MQQueue` para isso.

```
public void PutForwardMessage(MQMessage message);  
public void PutForwardMessage(MQMessage message, MQPutMessageOptions  
putMessageOptions);
```

Lança `MQException`.

Coloque uma mensagem sendo encaminhada para a fila, em que `message` é a mensagem original.

### **mensagem**

Um objeto `MQMessage` que contém os dados do descritor de mensagens e a mensagem a ser enviada. O descritor de mensagens pode ser alterado como uma consequência deste método. Os valores no descritor de mensagens imediatamente após a conclusão desse método são os valores que foram colocados na fila ou publicados no tópico.

Os seguintes códigos de razão são retornados para um cliente reconectável:

- `MQRC_CALL_INTERRUPTED` se a conexão for interrompida ao executar uma chamada `Put` em uma mensagem persistente e a reconexão for bem-sucedida.
- `MQRC_NONE` se a conexão for bem-sucedida ao executar uma chamada `Put` em uma mensagem não persistente (consulte [Recuperação de Aplicativo](#)).

### **putMessageOptions**

Opções que controlam a ação da entrada.

Se *putMessageOptions* não for especificado, a instância padrão de MQPutMessageOptions será usada

Se você usar a opção MQPMO\_LOGICAL\_ORDER em um cliente reconectável, o código de razão MQRC\_RECONNECT\_INCOMPATIBLE será retornado

```
public void PutReplyMessage(MQMessage message)  
public void PutReplyMessage(MQMessage message, MQPutMessageOptions  
putMessageOptions)
```

Lança MQException..

Coloque uma mensagem de resposta para a fila, em que *message* é a mensagem original

### **mensagem**

Contém o descritor de mensagens e os dados da mensagem retornados Alguns dos campos no descritor de mensagem são parâmetros de entrada É importante assegurar que os parâmetros de entrada MessageId e CorrelationId sejam configurados conforme necessário

Um cliente reconectável retorna o código de razão MQRC\_BACKED\_OUT após a reconexão bem-sucedida para mensagens recebidas em MQGM\_SYNCPOINT.

### **putMessageOptions**

Opções que controlam a ação da entrada.

Se *putMessageOptions* não for especificado, a instância padrão de MQPutMessageOptions será usada

Se você usar a opção MQPMO\_LOGICAL\_ORDER em um cliente reconectável, o código de razão MQRC\_RECONNECT\_INCOMPATIBLE será retornado

```
public void PutReportMessage(MQMessage message)  
public void PutReportMessage(MQMessage message, MQPutMessageOptions  
putMessageOptions)
```

Lança MQException..

Coloque uma mensagem de relatório sobre a fila, em que *message* é a mensagem original

### **mensagem**

Contém o descritor de mensagens e os dados da mensagem retornados Alguns dos campos no descritor de mensagem são parâmetros de entrada É importante assegurar que os parâmetros de entrada MessageId e CorrelationId sejam configurados conforme necessário

Um cliente reconectável retorna o código de razão MQRC\_BACKED\_OUT após a reconexão bem-sucedida para mensagens recebidas em MQGM\_SYNCPOINT.

### **putMessageOptions**

Opções que controlam a ação da entrada.

Se *putMessageOptions* não for especificado, a instância padrão de MQPutMessageOptions será usada

Se você usar a opção MQPMO\_LOGICAL\_ORDER em um cliente reconectável, o código de razão MQRC\_RECONNECT\_INCOMPATIBLE será retornado

## **Construtores**

```
public MQQueue MQQueueManager.AccessQueue(string queueName, int openOptions);  
public MQQueue MQQueueManager.AccessQueue(string queueName, int openOptions,  
string queueManagerName, string dynamicQueueName, string alternateUserId);
```

Lança MQException..

Acessa uma fila neste gerenciador.



É possível obter ou procurar mensagens, colocar mensagens, consultar sobre os atributos da fila ou configurar os atributos da fila. Se a fila denominada for uma fila modelo, uma fila local dinâmica será criada. Consulte o atributo name do objeto MQQueue resultante para descobrir o nome da fila dinâmica..

**queueName**

Nome da fila a abrir.

**openOptions**

Opções que controlam a abertura da fila.

**MQC.MQOO\_ALTERNATE\_USER\_AUTHORITY**

Validar com o identificador de usuários especificado.

**MQC.MQOO\_BIND\_AS\_QDEF**

Usar ligação padrão para a fila.

**MQC.MQOO\_BIND\_NOT\_FIXED**

Não ligar com um destino específico.

**MQC.MQOO\_BIND\_ON\_OPEN**

Ligar identificador com destino quando a fila for aberta.

**MQC.MQOO\_BROWSE**

Abrir para navegar pela mensagem.

**MQC.MQOO\_FAIL\_IF QUIESCING**

Falha se o gerenciador de filas estiver inativo.

**MQC.MQOO\_INPUT\_AS\_Q\_DEF**

Abrir para obter mensagens usando padrão definido pela fila.

**MQC.MQOO\_INPUT\_SHARED**

Abrir para obter mensagens com acesso compartilhado.

**MQC.MQOO\_INPUT\_EXCLUSIVE**

Abrir para obter mensagens com acesso exclusivo.

**MQC.MQOO\_INQUIRE**

Abrir para consulta - necessário se você deseja consultar propriedades.

**MQC.MQOO\_OUTPUT**

Abrir para colocar mensagens.

**MQC.MQOO\_PASS\_ALL\_CONTEXT**

Permitir que todo contexto seja transmitido.

**MQC.MQOO\_PASS\_IDENTITY\_CONTEXT**

Permitir que contexto de identidade seja transmitido.

**MQC.MQOO\_SAVE\_ALL\_CONTEXT**

Salvar o contexto quando a mensagem for recuperada

**MQC.MQOO\_SET**

Abrir para configurar atributos-necessário se você deseja configurar propriedades

**MQC.MQOO\_SET\_ALL\_CONTEXT**

Permite que todo contexto seja configurado.

**MQC.MQOO\_SET\_IDENTITY\_CONTEXT**

Permite que o contexto de identidade seja configurado.

**queueManagerName**

Nome do gerenciador de filas no qual a fila é definida. Um nome inteiramente em branco ou nulo denota o gerenciador de filas ao qual o objeto MQQueueManager está conectado.

**DynamicQueueName**

*dynamicQueueName* é ignorado, a menos que queueName especifique o nome de uma fila modelo. Se sim, *dynamicQueueName* especifica o nome da fila dinâmica a ser criada. Um nome em branco ou nulo não será válido se queueName especificar o nome de uma fila modelo. Se o último caractere não em branco no nome for um asterisco, \*, o gerenciador de filas substituirá o

asterisco por uma sequência de caracteres. Os caracteres garantem que o nome gerado para a fila seja exclusivo neste gerenciador.

#### **alternateUserId**

Se MQC.MQOO\_ALTERNATE\_USER\_AUTHORITY for especificado no parâmetro openOptions , *alternateUserId* especificará o identificador de usuário alternativo usado para verificar a autorização para a abertura. Se MQC.MQOO\_ALTERNATE\_USER\_AUTHORITY não for especificado, *alternateUserId* poderá ser deixado em branco ou nulo.

```
public MQQueue(MQQueueManager queueManager, string queueName, int openOptions,  
string queueManagerName, string dynamicQueueName, string alternateUserId);
```

Lança MQException..

Acessa uma fila em queueManager

É possível obter ou procurar mensagens, colocar mensagens, consultar sobre os atributos da fila ou configurar os atributos da fila. Se a fila denominada for uma fila modelo, uma fila local dinâmica será criada. Consulte o atributo name do objeto MQQueue resultante para descobrir o nome da fila dinâmica..

#### **queueManager**

O gerenciador de filas no qual acessar a fila.

#### **queueName**

Nome da fila a abrir.

#### **openOptions**

Opções que controlam a abertura da fila.

##### **MQC.MQOO\_ALTERNATE\_USER\_AUTHORITY**

Validar com o identificador de usuários especificado.

##### **MQC.MQOO\_BIND\_AS\_QDEF**

Usar ligação padrão para a fila.

##### **MQC.MQOO\_BIND\_NOT\_FIXED**

Não ligar com um destino específico.

##### **MQC.MQOO\_BIND\_ON\_OPEN**

Ligar identificador com destino quando a fila for aberta.

##### **MQC.MQOO\_BROWSE**

Abrir para navegar pela mensagem.

##### **MQC.MQOO\_FAIL\_IF QUIESCING**

Falha se o gerenciador de filas estiver inativo.

##### **MQC.MQOO\_INPUT\_AS\_Q\_DEF**

Abrir para obter mensagens usando padrão definido pela fila.

##### **MQC.MQOO\_INPUT\_SHARED**

Abrir para obter mensagens com acesso compartilhado.

##### **MQC.MQOO\_INPUT\_EXCLUSIVE**

Abrir para obter mensagens com acesso exclusivo.

##### **MQC.MQOO\_INQUIRE**

Abrir para consulta - necessário se você desejar consultar propriedades.

##### **MQC.MQOO\_OUTPUT**

Abrir para colocar mensagens.

##### **MQC.MQOO\_PASS\_ALL\_CONTEXT**

Permitir que todo contexto seja transmitido.

##### **MQC.MQOO\_PASS\_IDENTITY\_CONTEXT**

Permitir que contexto de identidade seja transmitido.

##### **MQC.MQOO\_SAVE\_ALL\_CONTEXT**

Salvar o contexto quando a mensagem for recuperada

### **MQC.MQOO\_SET**

Abrir para configurar atributos-necessário se você desejar configurar propriedades

### **MQC.MQOO\_SET\_ALL\_CONTEXT**

Permite que todo contexto seja configurado.

### **MQC.MQOO\_SET\_IDENTITY\_CONTEXT**

Permite que o contexto de identidade seja configurado.

### **queueManagerName**

Nome do gerenciador de filas no qual a fila é definida. Um nome inteiramente em branco ou nulo denota o gerenciador de filas ao qual o objeto MQQueueManager está conectado.

### **DynamicQueueName**

*dynamicQueueName* é ignorado, a menos que *queueName* especifique o nome de uma fila modelo. Se sim, *dynamicQueueName* especifica o nome da fila dinâmica a ser criada. Um nome em branco ou nulo não será válido se *queueName* especificar o nome de uma fila modelo. Se o último caractere não em branco no nome for um asterisco, \*, o gerenciador de filas substituirá o asterisco por uma sequência de caracteres. Os caracteres garantem que o nome gerado para a fila seja exclusivo neste gerenciador.

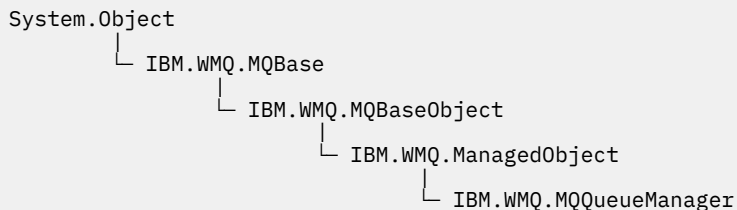
### **alternateUserId**

Se MQC.MQOO\_ALTERNATE\_USER\_AUTHORITY for especificado no parâmetro *openOptions*, *alternateUserId* especificará o identificador de usuário alternativo usado para verificar a autorização para a abertura. Se MQC.MQOO\_ALTERNATE\_USER\_AUTHORITY não for especificado, *alternateUserId* poderá ser deixado em branco ou nulo.

## **Classe MQQueueManager.NET**

Use MQQueueManager para se conectar a um gerenciador de filas e acessar objetos do gerenciador de filas. Também controla transações. O construtor MQQueueManager cria uma conexão cliente ou servidor.

### **Parâmetro class**



```
public class IBM.WMQ.MQQueueManager extends IBM.WMQ.MQManagedObject;
```

- [“Propriedades” na página 1823](#)
- [“Métodos” na página 1827](#)
- [“Construtores” na página 1832](#)

### **Propriedades**

Teste para MQException ser lançado ao obter propriedades.

```
public int AccountingConnOverride {get;}
```

Se os aplicativos podem substituir a configuração dos valores de contabilidade de MQI e de contabilidade de fila.

```
public int AccountingInterval {get;}
```

Quanto tempo antes dos registros de conta intermediários serem gravados (em segundos).

```
public int ActivityRecording {get;}
```

Controla a geração de relatório de atividades.

**public int AdoptNewMCACheck {get;}**

Especifica quais elementos são verificados para determinar se o MCA é adotado quando um novo canal de entrada é detectado.. Para ser adotado, o nome do MCA deve corresponder ao nome de um MCA ativo

**public int AdoptNewMCAInterval {get;}**

A quantidade de tempo, em segundos, que o novo canal aguarda pela finalização do canal órfão.

**public int AdoptNewMCAType {get;}**

Se uma instância MCA órfã deve ser adotada (reiniciada) quando um novo pedido de canal de entrada for detectado correspondente ao valor de AdoptNewMCACheck.

**public int BridgeEvent {get;}**

Se eventos de ponte IMS são gerados.

**public int ChannelEvent {get;}**

Se os eventos do canal são gerados.

**public int ChannelInitiatorControl {get;}**

Se o inicializador de canais inicia automaticamente quando o gerenciador de filas é iniciado.

**public int ChannelInitiatorAdapters {get;}**

O número de subtarefas do adaptador para processar chamadas IBM MQ .

**public int ChannelInitiatorDispatchers {get;}**

O número de dispatchers para usar para o inicializador de canais.

**public int ChannelInitiatorTraceAutoStart {get;}**

Especifica se o rastreamento do inicializador de canais inicia automaticamente.

**public int ChannelInitiatorTraceTableSize {get;}**

O tamanho, em megabytes, do espaço de dados de rastreamento de um inicializador de canais

**public int ChannelMonitoring {get;}**

Se o monitoramento de canal é usado.

**public int ChannelStatistics {get;}**

Controla a coleção de dados estatísticos para canais.

**public int CharacterSet {get;}**

Retorna o identificador do conjunto de caracteres codificados (CCSID) do Gerenciador de Filas.

CharacterSet é usado pelo gerenciador de filas para todos os campos de sequência de caractere na interface de programação do aplicativo

**public int ClusterSenderMonitoring {get;}**

Controla a coleção de dados de monitoramento on-line para canais do emissor de cluster definidos automaticamente

**public int ClusterSenderStatistics {get;}**

Controla a coleta de dados estatísticos para canais emissores de cluster definidos automaticamente.

**public int ClusterWorkLoadMRU {get;}**

O número máximo de canais de cluster de saída.

**public int ClusterWorkLoadUseQ {get;}**

O valor padrão da propriedade MQQueue , ClusterWorkLoadUseQ, se especificar um valor de QMGR.

**public int CommandEvent {get;}**

Especifica se eventos do comando são gerados.

**public string CommandInputQueueName {get;}**

Retorna o nome da fila de entrada de comando definida no gerenciador de filas. Os aplicativos podem enviar comandos para essa fila, se autorizados a fazê-lo.

**public int CommandLevel {get;}**

Indica o nível de função do gerenciador de filas. O conjunto de funções que correspondem a um nível de função particular depende da plataforma. Em uma plataforma específica, é possível contar com cada gerenciador de filas que suporte as funções no nível funcional mais baixo comum para todos os gerenciadores de filas

**public int CommandLevel {get;}**

Se o servidor de comandos inicia automaticamente quando o gerenciador de filas é iniciado.

**public string DNSGroup {get;}**

Não é mais usado.

**public int DNSWLM {get;}**

Não é mais usado.

**public int IPAddressVersion {get;}**

Qual protocolo IP (IPv4 ou IPv6) usar para uma conexão de canal.

**public boolean IsConnected {get;}**

Retorna o valor do `isConnected`.

Se `true`, uma conexão com o gerenciador de filas foi feita e não é conhecida como interrompida..

Quaisquer chamadas para `IsConnected` não tentam ativamente alcançar o gerenciador de fila, portanto, é possível que a conectividade física possa quebrar, mas `IsConnected` ainda pode retornar `true`. O estado `IsConnected` é atualizado apenas quando a atividade, por exemplo: colocar uma mensagem, obter uma mensagem, é executada no gerenciador de filas

Se `false`, uma conexão com o gerenciador de filas não foi feita, ou foi interrompida ou foi desconectada

**public int KeepAlive {get;}**

Especifica se o recurso TCP `KEEPALIVE` deve ser usado para verificar se a outra extremidade da conexão ainda está disponível. Se estiver indisponível, o canal será fechado.

**public int ListenerTimer {get;}**

O intervalo de tempo, em segundos, entre as tentativas do IBM MQ de reiniciar o listener após uma falha de APPC ou TCP/IP..

**public int LoggerEvent {get;}**

Se eventos do criador de logs são gerados.

**public string LU62ARMSuffix {get;}**

O sufixo do membro APPCPM de `SYS1.PARMLIB`. Esse sufixo nomeia o `LUADD` para esse iniciador de canal. Quando o gerenciador de reinício automático (ARM) reiniciar o inicializador de canais, o comando `z/OS SET APPC=xx` será emitido.

**public string LUGroupName {get; z/os}**

O nome da LU genérica a ser usado pelo listener da LU 6.2 que manipula transmissões de entrada para o grupo de filas compartilhadas..

**public string LUName {get;}**

O nome da LU a usar para transmissões de LU 6.2 de saída.

**public int MaximumActiveChannels {get;}**

O número máximo de canais que podem estar ativos a qualquer momento.

**public int MaximumCurrentChannels {get;}**

O número máximo de canais que podem ser atuais a qualquer momento (incluindo canais de conexão do servidor com clientes conectados).

**public int MaximumLU62Channels {get;}**

O número máximo de canais que podem ser atuais ou clientes que podem ser conectados, que usam o protocolo de transmissão de LU 6.2.

**public int MaximumMessageLength {get;}**

Retorna o comprimento máximo de uma mensagem (em bytes) que pode ser tratado pelo gerenciador de filas. Nenhuma fila pode ser definida com um comprimento máximo de mensagem maior do que `MaximumMessageLength`

**public int MaximumPriority {get;}**

Retorna a prioridade da mensagem máxima suportada pelo gerenciador de filas. As prioridades variam de zero (mais baixa) até este valor. Lança `MQException` se você chamar esse método após desconectar do gerenciador de filas.

**public int MaximumTCPChannels {get;}**

O número máximo de canais que podem ser atuais ou clientes que podem ser conectados, que usam o protocolo de transmissão TCP/IP.

**public int MQIAccounting {get;}**

Controla a coleção de dados da conta para dados do MQI.

**public int MQIStatistics {get;}**

Controla a coleção de informações de monitoramento de estatísticas para o gerenciador de filas.

**public int OutboundPortMax {get;}**

O valor máximo no intervalo de números de porta a ser usado ao ligar canais de saída.

**public int OutboundPortMin {get;}**

O valor mínimo no intervalo de números de porta a ser usado ao ligar canais de saída.

**public int QueueAccounting {get;}**

Se a contabilidade de classe 3 (contabilidade de nível de encadeamento e de nível de fila) deve ser usada para todas as filas.

**public int QueueMonitoring {get;}**

Controla a coleção de dados de monitoramento on-line para filas.

**public int QueueStatistics {get;}**

Controla a coleção de dados estatísticos para filas.

**public int ReceiveTimeout {get;}**

O período de tempo que um canal TCP/IP aguarda para receber dados, incluindo pulsações, de seu parceiro antes de retornar para o estado inativo.

**public int ReceiveTimeoutMin {get;}**

O período mínimo de tempo que um canal TCP/IP aguarda para receber dados, incluindo pulsações, de seu parceiro antes de retornar para um estado inativo.

**public int ReceiveTimeoutType {get;}**

O qualificador para aplicar ao valor em ReceiveTimeout.

**public int SharedQueueQueueManagerName {get;}**

Especifica como entregar mensagens para uma fila compartilhada.. Se o put especificar um gerenciador de filas diferente do mesmo grupo de filas compartilhadas que o gerenciador de filas de destino, a mensagem será entregue de duas maneiras:

**MQC.MQSQQM\_USE**

As mensagens são entregues para o gerenciador de fila de objetos antes de serem colocadas na fila compartilhada

**MQCMQSQQM\_IGNORE**

As mensagens são colocadas diretamente na fila compartilhada

**public int SSLEvent {get;}**

Se eventos TLS são gerados.

**public int SSLFips {get;}**

Se apenas algoritmos certificados pelo FIPS devem ser usados se a criptografia for executada no IBM MQ, em vez do hardware criptográfico

**public int SSLKeyResetCount {get;}**

Indica o número de bytes não criptografados enviados e recebidos dentro de uma conversa TLS antes que a chave secreta seja renegociada

**public int ClusterSenderStatistics {get;}**

Especifica o intervalo, em minutos, entre coletas consecutivas de estatísticas.

**public int SyncpointAvailability {get;}**

Indica se o gerenciador de filas suporta unidades de trabalho e pontos de sincronização com os métodos MQQueue.get e MQQueue.put .

**public string TCPName {get;}**

O nome do sistema TCP/IP único ou padrão a ser usado, dependendo do valor de TCPStackType.

**public int TCPStackType {get;}**

Especifica se o inicializador de canais usa apenas o espaço de endereço TCP/IP especificado em TCPName. Como alternativa, o inicializador de canais pode se ligar a qualquer endereço TCP/IP..

**public int TraceRouteRecording {get;}**

Controla a gravação das informações de rastreamento da rota.

## Métodos

**public MQProcess AccessProcess(string processName, int openOptions);**

**public MQProcess AccessProcess(string processName, int openOptions, string queueManagerName, string alternateUserId);**

Lança MQException..

Acesse um processo IBM MQ neste gerenciador de filas para consultar atributos de processo.

### processName

O nome do processo a abrir.

### openOptions

Opções que controlam a abertura do processo. As opções válidas que podem ser incluídas ou combinadas usando um OR bit a bit são:

- MQC.MQOO\_FAIL\_IF QUIESCING
- MQC.MQOO\_INQUIRE
- MQC.MQOO\_SET
- MQC.MQOO\_ALTERNATE\_USER\_AUTHORITY

### queueManagerName

O nome do gerenciador de filas no qual o processo está definido. É possível deixar um nome de gerenciador de fila em branco ou nulo se o gerenciador de filas for igual ao que o processo está acessando.

### alternateUserId

Se MQC.MQOO\_ALTERNATE\_USER\_AUTHORITY for especificado no parâmetro **openOptions**, *alternateUserId* especificará o ID do usuário alternativo usado para verificar a autorização para a ação. Se MQOO\_ALTERNATE\_USER\_AUTHORITY não for especificado, *alternateUserId* poderá ficar em branco ou nulo.

A autoridade do usuário padrão será usada para a conexão com o gerenciador de fila se MQC.MQOO\_ALTERNATE\_USER\_AUTHORITY não for especificado

**public MQQueue AccessQueue(string queueName, int openOptions);**

**public MQQueue AccessQueue(string queueName, int openOptions, string queueManagerName, string dynamicQueueName, string alternateUserId);**

Lança MQException..

Acessa uma fila neste gerenciador.

É possível obter ou procurar mensagens, colocar mensagens, consultar sobre os atributos da fila ou configurar os atributos da fila. Se a fila denominada for uma fila modelo, uma fila local dinâmica será criada. Consulte o atributo name do objeto MQQueue resultante para descobrir o nome da fila dinâmica..

### queueName

Nome da fila a abrir.

### openOptions

Opções que controlam a abertura da fila.

#### **MQC.MQOO\_ALTERNATE\_USER\_AUTHORITY**

Validar com o identificador de usuários especificado.

**MQC.MQOO\_BIND\_AS\_QDEF**

Usar ligação padrão para a fila.

**MQC.MQOO\_BIND\_NOT\_FIXED**

Não ligar com um destino específico.

**MQC.MQOO\_BIND\_ON\_OPEN**

Ligar identificador com destino quando a fila for aberta.

**MQC.MQOO\_BROWSE**

Abrir para navegar pela mensagem.

**MQC.MQOO\_FAIL\_IF QUIESCING**

Falha se o gerenciador de filas estiver inativo.

**MQC.MQOO\_INPUT\_AS\_Q\_DEF**

Abrir para obter mensagens usando padrão definido pela fila.

**MQC.MQOO\_INPUT\_SHARED**

Abrir para obter mensagens com acesso compartilhado.

**MQC.MQOO\_INPUT\_EXCLUSIVE**

Abrir para obter mensagens com acesso exclusivo.

**MQC.MQOO\_INQUIRE**

Abrir para consulta - necessário se você desejar consultar propriedades.

**MQC.MQOO\_OUTPUT**

Abrir para colocar mensagens.

**MQC.MQOO\_PASS\_ALL\_CONTEXT**

Permitir que todo contexto seja transmitido.

**MQC.MQOO\_PASS\_IDENTITY\_CONTEXT**

Permitir que contexto de identidade seja transmitido.

**MQC.MQOO\_SAVE\_ALL\_CONTEXT**

Salvar o contexto quando a mensagem for recuperada

**MQC.MQOO\_SET**

Abrir para configurar atributos-necessário se você desejar configurar propriedades

**MQC.MQOO\_SET\_ALL\_CONTEXT**

Permite que todo contexto seja configurado.

**MQC.MQOO\_SET\_IDENTITY\_CONTEXT**

Permite que o contexto de identidade seja configurado.

**queueManagerName**

Nome do gerenciador de filas no qual a fila é definida. Um nome inteiramente em branco ou nulo denota o gerenciador de filas ao qual o objeto MQQueueManager está conectado.

**DynamicQueueName**

*dynamicQueueName* é ignorado, a menos que *queueName* especifique o nome de uma fila modelo. Se sim, *dynamicQueueName* especifica o nome da fila dinâmica a ser criada. Um nome em branco ou nulo não será válido se *queueName* especificar o nome de uma fila modelo. Se o último caractere não em branco no nome for um asterisco, \*, o gerenciador de filas substituirá o asterisco por uma sequência de caracteres. Os caracteres garantem que o nome gerado para a fila seja exclusivo neste gerenciador.

**alternateUserId**

Se MQC.MQOO\_ALTERNATE\_USER\_AUTHORITY for especificado no parâmetro *openOptions*, *alternateUserId* especificará o identificador de usuário alternativo usado para verificar a autorização para a abertura. Se MQC.MQOO\_ALTERNATE\_USER\_AUTHORITY não for especificado, *alternateUserId* poderá ser deixado em branco ou nulo.



```

public MQTopic AccessTopic( MQDestination destination, string topicName, string
topicObject, int options);
public MQTopic AccessTopic( MQDestination destination, string topicName, string
topicObject, int options, string alternateUserId);
public MQTopic AccessTopic( MQDestination destination, string topicName, string
topicObject, int options, string alternateUserId, string subscriptionName);
public MQTopic AccessTopic( MQDestination destination, string topicName, string
topicObject, int options, string alternateUserId, string subscriptionName,
System.Collections.Hashtable properties);
public MQTopic AccessTopic(string topicName, string topicObject, int openAs,
int options);
public MQTopic AccessTopic(string topicName, string topicObject, int openAs,
int options, string alternateUserId);
public MQTopic AccessTopic(string topicName, string topicObject, int options,
string alternateUserId, string subscriptionName);
public MQTopic AccessTopic(string topicName, string topicObject, int options,
string alternateUserId, string subscriptionName, System.Collections.Hashtable
properties);

```

Acesse um tópico neste gerenciador de filas

Os objetos MQTopic estão intimamente relacionados aos objetos do tópico administrativo, que às vezes são chamados de objetos do tópico Na entrada, o topicObject aponta para um objeto do tópico administrativo O construtor MQTopic obtém uma sequência de tópicos do objeto do tópico e a combina com topicName para criar um nome do tópico. topicObject ou topicName pode ser nulo. O nome do tópico é correspondido com a árvore de tópicos e o nome do objeto do tópico administrativo correspondente mais próximo é retornado em topicObject

Os tópicos associados ao objeto MQTopic são o resultado da combinação de duas sequências de tópicos. A primeira sequência de tópicos é definida pelo objeto do tópico administrativo identificado por topicObject.. A segunda sequência de tópicos é topicString A sequência de tópicos resultante associada com o objeto MQTopic pode identificar vários tópicos incluindo curingas

Dependendo se o tópico é aberto para publicação ou assinatura, é possível usar os métodos MQTopic .Put para publicar em tópicos ou métodos MQTopic .Get para receber publicações em tópicos.. Se você deseja publicar e assinar o mesmo tópico, deverá acessar o tópico duas vezes, uma vez para publicação e uma vez para assinatura.

Se você criar um objeto MQTopic para assinatura, sem fornecer um objeto MQDestination , uma assinatura gerenciada será assumida. Se você passar uma fila como um objeto MQDestination , uma assinatura não gerenciada será assumida Deve-se assegurar que as opções de assinatura configuradas sejam consistentes com a assinatura que está sendo gerenciada ou não gerenciada.

#### **destino**

*destination* é uma instância MQQueue . Ao fornecer *destination*, MQTopic é aberto como uma assinatura não gerenciada. As publicações sobre o tópico são entregues à fila acessada como *destination*.

#### **topicName**

Uma sequência de tópicos que é a segunda parte do nome do tópico.. O *topicName* é concatenado com a sequência de tópicos definida no objeto do tópico administrativo *topicObject* É possível configurar *topicName* como nulo, em cujo caso o nome do tópico é definido pela sequência de tópicos em *topicObject*.

#### **topicObject**

Na entrada, *topicObject* é o nome do objeto do tópico que contém a sequência de tópicos que forma a primeira parte do nome do tópico A sequência de tópicos em *topicObject* é concatenada com *topicName* As regras para construir sequências de tópicos são definidas em Combinando sequências de tópicos.

Na saída, *topicObject* contém o nome do objeto do tópico administrativo que é a correspondência mais próxima na árvore de tópicos para o tópico identificado pela sequência de tópicos

## openAs

Acesse o tópico para publicar ou assinar.. O parâmetro pode conter apenas uma dessas opções:

- MQC.MQTOPIC\_OPEN\_AS\_SUBSCRIPTION
- MQC.MQTOPIC\_OPEN\_AS\_PUBLICATION

## opções

Combine as opções que controlam a abertura do tópico para publicação ou assinatura. Use constantes MQC.MQSO\_\* para acessar um tópico para assinatura e constantes MQC.MQOO\_\* para acessar um tópico para publicação.

Se mais de uma opção for necessária, inclua os valores juntos ou combine os valores de opção usando o operador OR bit a bit

## alternateUserId

Especifique o ID do usuário alternativo usado para verificar a autorização necessária para concluir a operação. Deve-se especificar *alternateUserId*, se MQC.MQOO\_ALTERNATE\_USER\_AUTHORITY ou MQC.MQSO\_ALTERNATE\_USER\_AUTHORITY estiver configurado no parâmetro de opções

## subscriptionName

*subscriptionName* será necessário se as opções MQC.MQSO\_DURABLE ou MQC.MQSO\_ALTER forem fornecidas.. Nos dois casos, MQTopic é implicitamente aberto para assinatura. Uma exceção será lançada se o MQC.MQSO\_DURABLE estiver configurado e a assinatura existir ou se MQC.MQSO\_ALTER estiver configurado e a assinatura não existir.

## propriedades

Configure qualquer uma das propriedades de assinatura especiais listadas usando uma tabela de hash Entradas especificadas na tabela hash são atualizadas com valores de saída. Entradas não são incluídas na tabela hash para relatar valores de saída.

- MQC.MQSUB\_PROP\_ALTERNATE\_SECURITY\_ID
- MQC.MQSUB\_PROP\_SUBSCRIPTION\_EXPIRY
- MQC.MQSUB\_PROP\_SUBSCRIPTION\_USER\_DATA
- MQC.MQSUB\_PROP\_SUBSCRIPTION\_CORRELATION\_ID
- MQC.MQSUB\_PROP\_PUBLICATION\_PRIORITY
- MQC.MQSUB\_PROP\_PUBLICATION\_ACCOUNTING\_TOKEN
- MQC.MQSUB\_PROP\_PUBLICATION\_APPLICATIONID\_DATA

## public MQAsyncStatus GetAsyncStatus();

Lança MQException

Retorna um objeto MQAsyncStatus , que representa a atividade assíncrona para a conexão do gerenciador de filas

## public void Backout();

Lança MQException..

Restauração de quaisquer mensagens que foram lidas ou gravadas dentro do ponto de sincronização desde o último ponto de sincronização

As mensagens que foram gravadas com o sinalizador MQC.MQPMO\_SYNCPOINT configurado são removidas das filas. As mensagens lidas com a sinalização MQC.MQGMO\_SYNCPOINT são restabelecidas nas filas das quais vieram. Se as mensagens forem persistentes, as alterações serão registradas em log.

Para clientes reconectáveis, o código de razão MQRC\_NONE é retornado para um cliente após a reconexão ser bem-sucedida..

## **public void Begin();**

Lança MQException..

Begin é suportado apenas no modo de ligação do servidor Ele inicia uma unidade de trabalho global

## **public void Commit();**

Lança MQException..

Confirmar quaisquer mensagens que foram lidas ou gravadas no ponto de sincronização desde o último ponto de sincronização.

As mensagens gravadas com o sinalizador MQC.MQPMO\_SYNCPOINT configurado são disponibilizadas para outros aplicativos. As mensagens recuperadas com o conjunto de sinalizadores MQC.MQGMO\_SYNCPOINT são excluídas Se as mensagens forem persistentes, as alterações serão registradas em log.

Os seguintes códigos de razão são retornados para um cliente reconectável:

- MQRC\_CALL\_INTERRUPTED se a conexão for perdida durante a execução da chamada de confirmação
- MQRC\_BACKED\_OUT se a chamada de confirmação for emitida após a reconexão

## **Disconnect();**

Lança MQException..

Feche a conexão com o gerenciador de filas Todos os objetos acessados neste gerenciador de fila não são mais acessíveis a este aplicativo Para acessar novamente os objetos, crie um objeto MQQueueManager ..

Geralmente, qualquer trabalho executado como parte de uma unidade de trabalho é confirmado.. No entanto, se a unidade de trabalho for gerenciada pelo .NET, a unidade de trabalho poderá ser retrocedida de volta.

```
public void Put(int type, string destinationName, MQMessage message);  
public void Put(int type, string destinationName, MQMessage message  
MQPutMessageOptions putMessageOptions);  
public void Put(int type, string destinationName, string queueManagerName,  
string topicString, MQMessage message);  
public void Put(string queueName, MQMessage message);  
public void Put(string queueName, MQMessage message, MQPutMessageOptions  
putMessageOptions);  
public void Put(string queueName, string queueManagerName, MQMessage message);  
public void Put(string queueName, string queueManagerName, MQMessage message,  
MQPutMessageOptions putMessageOptions);  
public void Put(string queueName, string queueManagerName, MQMessage message,  
MQPutMessageOptions putMessageOptions, string alternateUserId);
```

Lança MQException..

Coloca uma única mensagem em uma fila ou tópico sem criar um objeto MQQueue ou MQTopic primeiro.

### **queueName**

O nome da fila na qual colocar a mensagem.

### **destinationName**

O nome de um objeto de destino.. É uma fila ou um tópico dependendo do valor de *type*.

### **Tipo**

O tipo de objeto de destino Você não deve combinar as opções.

### **MQC.MQOT\_Q**

Fila

## **MQC.MQOT\_TOPIC**

Tópico

### **queueManagerName**

O nome do gerenciador de filas ou do alias do gerenciador de filas, no qual a fila está definida. Se o tipo MQC.MQOT\_TOPIC for especificado, esse parâmetro será ignorado.

Se a fila for uma fila modelo, e o nome do gerenciador de filas resolvido não for esse gerenciador de fila, um MQException será lançado.

### **topicString**

*topicString* é combinado com o nome do tópico no objeto do tópico *destinationName* ..

*topicString* será ignorado se *destinationName* for uma fila..

### **mensagem**

A mensagem a enviar. A mensagem é um objeto de entrada/saída.

Os seguintes códigos de razão são retornados para um cliente reconectável:

- MQRC\_CALL\_INTERRUPTED se a conexão for interrompida ao executar uma chamada Put em uma mensagem persistente.
- MQRC\_NONE se a conexão for bem-sucedida ao executar uma chamada Put em uma mensagem não persistentes (consulte [Recuperação do Aplicativo](#)).

### **putMessageOptions**

Opções que controlam as ações da entrada.

Se você omitir *putMessageOptions*, uma instância padrão de *putMessageOptions* será criada. *putMessageOptions* é um objeto de entrada / saída.

Se você usar a opção MQPMO\_LOGICAL\_ORDER em um cliente reconectável, o código de razão MQRC\_RECONNECT\_INCOMPATIBLE será retornado.

### **alternateUserId**

Especifica um identificador de usuário alternativo utilizado para verificar a autorização ao colocar a mensagem em uma fila.

É possível omitir *alternateUserId* se você não configurar MQC.MQOO\_ALTERNATE\_USER\_AUTHORITY em *putMessageOptions*. Se você configurar MQC.MQOO\_ALTERNATE\_USER\_AUTHORITY, também deverá configurar *alternateUserId*. *alternateUserId* não tem efeito a menos que você também configure MQC.MQOO\_ALTERNATE\_USER\_AUTHORITY.

## **Construtores**

```
public MQQueueManager();  
public MQQueueManager(string queueManagerName);  
public MQQueueManager(string queueManagerName, Int options);  
public MQQueueManager(string queueManagerName, Int options, string channel,  
string connName);  
public MQQueueManager(string queueManagerName, string channel, string  
connName);  
public MQQueueManager(string queueManagerName, System.Collections.Hashtable  
properties);
```

Lança MQException..

Cria uma conexão para um gerenciador de filas. Selecione entre criar uma conexão do cliente ou uma conexão do servidor.

Deve-se ter autoridade de consulta (inq) no gerenciador de filas ao tentar se conectar ao gerenciador de filas. Sem autoridade de consulta, a tentativa de conexão falhará.

Uma conexão do cliente será criada se uma das seguintes condições for verdadeira:

1. *channel* ou *connName* são especificados no construtor
2. *HostName*, *Port* ou *Channel* são especificados em *properties*..
3. *MQEnvironment.HostName*, *MQEnvironment.Port* ou *MQEnvironment.Channel* são especificados..

Os valores das propriedades de conexão são padronizadas na ordem mostrada. O *channel* e o *connName* no construtor têm precedência sobre os valores de propriedades no construtor. Os valores da propriedade do construtor têm precedência sobre as propriedades *MQEnvironment*.

O nome do host, nome do canal e porta são definidos na classe *MQEnvironment*.

### **queueManagerName**

Nome do gerenciador de filas ou do grupo de gerenciadores de fila ao qual se conectar

Omita o parâmetro ou deixe-o nulo ou em branco para fazer uma seleção do gerenciador de filas padrão. A conexão do gerenciador de filas padrão em um servidor é para o gerenciador de fila padrão no servidor. A conexão do gerenciador de filas padrão em uma conexão do cliente é para o gerenciador de filas ao qual o listener está conectado.

### **opções**

Especifique opções de conexão *MQCNO*. Os valores devem ser aplicáveis ao tipo de conexão sendo feita. Por exemplo, se você especificar as seguintes propriedades de conexão do servidor para uma conexão do cliente, *MQException* será lançado.

- *MQC.MQCNO\_FASTPATH\_BINDING*
- *MQC.MQCNO\_STANDARD\_BINDING*

### **propriedades**

O parâmetro de propriedades usa uma série de pares chave / valor que substituem as propriedades configuradas por *MQEnvironment*; Consulte o exemplo, [“Substituir propriedades MQEnvironment”](#) na página 1835. As seguintes propriedades podem ser substituídas:

- *MQC.CONNECT\_OPTIONS\_PROPERTY*
- *MQC.CONNECTION\_NAME\_PROPERTY*
- *MQC.ENCRYPTION\_POLICY\_SUITE\_B*
- *MQC.HOST\_NAME\_PROPERTY*
- *MQC.PORT\_PROPERTY*
- *MQC.CHANNEL\_PROPERTY*
- *MQC.SSL\_CIPHER\_SPEC\_PROPERTY*
- *MQC.SSL\_PEER\_NAME\_PROPERTY*
- *MQC.SSL\_CERT\_STORE\_PROPERTY*
- *MQC.SSL\_CRYPTO\_HARDWARE\_PROPERTY*
- *MQC.SECURITY\_EXIT\_PROPERTY*
- *MQC.SECURITY\_USERDATA\_PROPERTY*
- *MQC.SEND\_EXIT\_PROPERTY*
- *MQC.SEND\_USERDATA\_PROPERTY*
- *MQC.RECEIVE\_EXIT\_PROPERTY*
- *MQC.RECEIVE\_USERDATA\_PROPERTY*
- *MQC.USER\_ID\_PROPERTY*
- *MQC.PASSWORD\_PROPERTY*
- *MQC.MQAIR\_ARRAY*
- *MQC.KEY\_RESET\_COUNT*

- MQC.FIPS\_REQUIRED
- MQC.HDR\_CMP\_LIST
- MQC.MSG\_CMP\_LIST
- MQC.TRANSPORT\_PROPERTY

#### channel

Nome de um canal de conexão do servidor

#### connName

Nome da conexão no formato *HostName (Porta)*.

É possível fornecer uma lista de *nomes do host e portas* como um argumento para o construtor `MQQueueManager` (`String queueManagerName, Hashtable properties`) usando `CONNECTION_NAME_PROPERTY`

Por exemplo:

```
ConnectionName = "fred.mq.com(2344),nick.mq.com(3746),tom.mq.com(4288)";
Hashtable Properties=new Hashtable();
properties.Add(MQC.CONNECTION_NAME_PROPERTY,ConnectionName);
MQQueueManager qmgr=new MQQueueManager("qmgrname",properties);
```

Quando uma tentativa de conexão é feita, a lista de nomes de conexão é processada em ordem... Se a tentativa de conexão com o primeiro nome do host e porta falhar, a conexão com o segundo par de atributos será tentada. O cliente repete este processo até que uma conexão bem-sucedida seja estabelecida ou a lista seja esgotada. Se a lista for esgotada, um código de razão e um código de conclusão apropriados serão retornados para o aplicativo cliente

Quando um número da porta não é fornecido para o nome da conexão, a porta padrão (configurada em `mqclient.ini`) é usado

## Configure a lista de conexões

É possível configurar a lista de conexões usando os métodos a seguir quando as opções de reconexão automática do cliente são configuradas:

### Configure a lista de conexões através de MQSERVER

É possível configurar a lista de conexão por meio do prompt de comandos

No prompt de comandos, configure o comando a seguir:

```
MQSERVER=SYSTEM.DEF.SVRCONN/TCP/Hostname1(Port1),Hostname2(Por2),Hostname3(Port3)
```

Por exemplo:

```
MQSERVER=SYSTEM.DEF.SVRCONN/TCP/fred.mq.com(5266),nick.mq.com(6566),jack.mq.com(8413)
```

Se configurar a conexão no MQSERVER, não a configure no aplicativo.

Se você configurar a lista de conexões no aplicativo, o aplicativo sobrescreverá tudo o que estiver configurado na variável de ambiente MQSERVER

### Configure a lista de conexões através do aplicativo

É possível configurar a lista de conexão no aplicativo, especificando o nome do host e as propriedades da porta

```
String connName = "fred.mq.com(2344), nick.mq.com(3746), chris.mq.com(4288)";
MQQueueManager qm = new MQQueueManager("QM1", "TestChannel", connName);
```

## Configure a lista de conexões por app.config

App.config é um arquivo XML no qual você especifica os pares chave-valor.

Na lista de conexões, especifique

```
<app.Settings>
<add key="Connection1" value="Hostname1(Port1)"/>
<add key="Connection2" value="Hostname2(Port2)"/>
</app.Settings>
```

Por exemplo:

```
<app.Settings>
<add key="Connection1" value="fred.mq.com(2966)"/>
<add key="Connection2" value="alex.mq.com(6533)"/>
</app.Settings>
```

É possível alterar diretamente a lista de conexões no arquivo app.config

## Configure a lista de conexões por MQEnvironment

Para configurar a lista Conexão por meio do MQEnvironment, use a propriedade *ConnectionName*.

```
MQEnvironment.ConnectionName = "fred.mq.com(4288),alex.mq.com(5211);
```

A propriedade *ConnectionName* sobrescreve o nome do host e as propriedades da porta configurados no MQEnvironment..

## Criar uma conexão de cliente

O exemplo a seguir mostra como criar uma conexão do cliente com um gerenciador de filas.. É possível criar uma conexão do cliente configurando as variáveis MQEnvironment antes de criar um novo Objeto MQQueueManager ..

```
MQEnvironment.Hostname = "fred.mq.com"; // host to connect to
MQEnvironment.Port     = 1414;          // port to connect to
                                   //If not explicitly set,
                                   // defaults to 1414
                                   // (the default IBM MQ port)
MQEnvironment.Channel  = "channel.name"; // the case sensitive
                                   // name of the
                                   // SVR CONN channel on
                                   // the queue manager
MQQueueManager qMgr    = new MQQueueManager("MYQM");
```

Figura 11. Conexão do cliente

## Substituir propriedades MQEnvironment

O exemplo a seguir mostra como criar um gerenciador de filas com seu ID do usuário e senha definidos em uma tabela hash.

```

Hashtable properties = new Hashtable();

properties.Add( MQC.USER_ID_PROPERTY, "ExampleUserId" );
properties.Add( MQC.PASSWORD_PROPERTY, "ExamplePassword" );

try
{
    MQQueueManager qMgr = new MQQueueManager("qmgrname", properties);
}
catch (MQException mqe)
{
    System.Console.WriteLine("Connect failed with " + mqe.Message);
    return((int)mqe.Reason);
}

```

Figura 12. Substituindo propriedades MQEnvironment

### Criar uma conexão reconectável

O exemplo a seguir mostra como reconectar automaticamente um cliente a um Gerenciador de Filas.

```

Hashtable properties = new Hashtable(); // The queue manager name and the
// properties how it has to be connected

properties.Add(MQC.CONNECT_OPTIONS_PROPERTY, MQC.MQCNO_RECONNECT); // Options
// through which reconnection happens

properties.Add(MQC.CONNECTION_NAME_PROPERTY, "fred.mq.com(4789),nick.mq.com(4790)"); // The list
// of queue managers through which reconnection happens

MQ QueueManager qmgr = new MQQueueManager("qmgrname", properties);

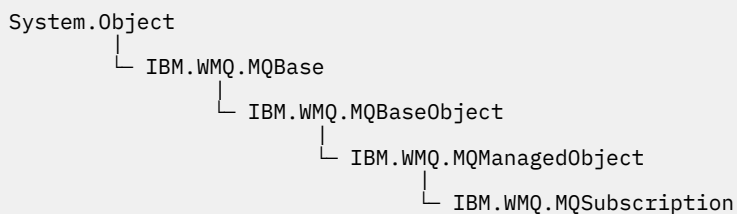
```

Figura 13. Reconectando automaticamente um cliente a um gerenciador de filas

## Classe MQSubscription.NET

Use MQSubscription para solicitar que as publicações retidas sejam enviadas ao assinante. MQSubscription é uma propriedade de um objeto MQTopic aberto para assinatura..

### Parâmetro class



```
public class IBM.WMQ.MQSubscription extends IBM.WMQ.MQManagedObject;
```

- [“Propriedades” na página 1836](#)
- [“Métodos” na página 1837](#)
- [“Construtores” na página 1837](#)

### Propriedades

Acesse as propriedades de assinatura usando a classe MQManagedObject ; consulte [“Propriedades” na página 1794](#)



## Métodos

Acesse os métodos `Inquire`, `Set` e `Get` de assinatura usando a classe `MQManagedObject` ; consulte [“Métodos”](#) na página 1795

```
public int RequestPublicationUpdate(int options);
```

Lança `MQException`..

Solicite uma publicação atualizada para o tópico atual Se o gerenciador de filas tiver uma publicação retida para o tópico, elas serão enviadas para o assinante

Antes de chamar `RequestPublicationUpdate`, abra um tópico para a assinatura para obter um objeto `MQSubscription`

Geralmente, abra a assinatura com a opção `MQC.MQSO_PUBLICATIONS_ON_REQUEST` . Se nenhum curinga estiver presente na sequência de tópicos, apenas uma publicação será enviada como resultado dessa chamada.. Se a sequência de tópicos contiver curingas, muitas publicações poderão ser enviadas O método retorna o número de publicações retidas enviadas para a fila de assinaturas. Não há garantia de que essas publicações sejam recebidas, especialmente se forem mensagens não persistentes.

### opções

#### **MQC.MQSRO\_FAIL\_IF QUIESCING**

O método falha se o gerenciador de filas estiver em um estado inativo. No z/OS, para um aplicativo CICS ou IMS , `MQC.MQSRO_FAIL_IF QUIESCING` também força o método a falhar se a conexão estiver em um estado quiesce.

#### **MQC.MQSRO\_NONE**

Nenhuma opção foi especificada

## Construtores

Nenhum construtor Público .

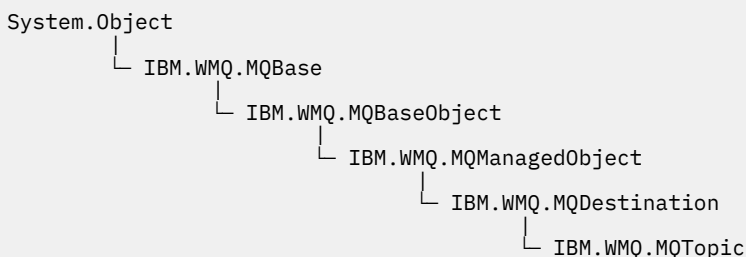
Um objeto `MQSubscription` é retornado na propriedade `SubscriptionReference` de um objeto `MQTopic` que é aberto para assinatura,

Chame o método `RequestPublicationUpdate` .. `MQSubscription` é uma subclasse de `MQManagedObject`.. Use a referência para acessar as propriedades e os métodos de `MQManagedObject`

## Classe MQTopic.NET

Use `MQTopic` para publicar ou assinar mensagens em um tópico ou consultar ou configurar atributos de um tópico. Crie um objeto `MQTopic` para publicação ou assinatura usando um construtor ou o método `MQQueueManager.AccessTopic` ..

### Parâmetro class



```
public class IBM.WMQ.MQTopic extends IBM.WMQ.MQDestination;
```

- [“Propriedades”](#) na página 1838

- “Métodos” na página 1838
- “Construtores” na página 1840

## Propriedades

Teste para `MQException` ser lançado ao obter propriedades.

### **public Boolean IsDurable {get;}**

Propriedade somente leitura que retorna `True` se a subscrição for durável ou `False` caso contrário. Se o tópico tiver sido aberto para publicação, a propriedade será ignorada e sempre retornará `False`.

### **public Boolean IsManaged {get;};**

Propriedade somente leitura que retorna `True` se a assinatura for gerenciada pelo gerenciador de filas ou `False` caso contrário. Se o tópico foi aberto para publicação, a propriedade será ignorada e sempre retornará `False`.

### **public Boolean IsSubscribed {get;};**

Propriedade somente leitura que retorna `True` se o tópico foi aberto para assinatura e `False` se o tópico foi aberto para publicação.

### **public MQSubscription SubscriptionReference {get;};**

A propriedade somente leitura que retorna o objeto `MQSubscription` associado a um objeto de tópico aberto para assinatura. A referência estará disponível se você desejar modificar as opções de fechamento ou iniciar qualquer um dos métodos de objetos.

### **public MQDestination UnmanagedDestinationReference {get;};**

Propriedade somente leitura que retorna o `MQQueue` associado a uma assinatura não gerenciada. Ele é o destino especificado quando o objeto do tópico foi criado. A propriedade retorna `null` para quaisquer objetos de tópico abertos para publicação ou com uma assinatura gerenciada.

## Métodos

### **public void Put(MQMessage message);**

### **public void Put(MQMessage message, MQPutMessageOptions putMessageOptions);**

Lança `MQException`.

Publica uma mensagem no tópico.

Modificações no objeto `MQMessage` após a chamada `Put` ter sido realizada não afetam a mensagem real na fila IBM MQ ou tópico de publicação.

`Put` atualiza as propriedades `MessageId` e `CorrelationId` do objeto `MQMessage` e não limpa dados da mensagem. Chamadas `Put` ou `Get` adicionais se referem às informações atualizadas no objeto `MQMessage`. Por exemplo, no fragmento de código a seguir, a primeira mensagem contém a e a segunda ab.

```
msg.WriteString("a");
q.Put(msg, pmo);
msg.WriteString("b");
q.Put(msg, pmo);
```

### **mensagem**

Um objeto `MQMessage` que contém os dados do descritor de mensagens e a mensagem a ser enviada. O descritor de mensagens pode ser alterado como uma consequência deste método. Os valores no descritor de mensagens imediatamente após a conclusão desse método são os valores que foram colocados na fila ou publicados no tópico.

Os seguintes códigos de razão são retornados para um cliente reconectável:

- `MQRC_CALL_INTERRUPTED` se a conexão for interrompida ao executar uma chamada `Put` em uma mensagem persistente e a reconexão for bem-sucedida.
- `MQRC_NONE` se a conexão for bem-sucedida ao executar uma chamada `Put` em uma mensagem não persistente (consulte [Recuperação de Aplicativo](#)).

### **putMessageOptions**

Opções que controlam a ação da entrada.

Se *putMessageOptions* não for especificado, a instância padrão de `MQPutMessageOptions` será usada

Se você usar a opção `MQPMO_LOGICAL_ORDER` em um cliente reconectável, o código de razão `MQRC_RECONNECT_INCOMPATIBLE` será retornado

**Nota:** Para simplicidade e desempenho, se você desejar colocar uma única mensagem em uma fila, use o objeto `MQQueueManager`. Deve-se ter um objeto `MQQueue` para isso

```
public void Get(MQMessage message);  
public void Get(MQMessage message, MQGetMessageOptions getMessageOptions);  
public void Get(MQMessage message, MQGetMessageOptions getMessageOptions, int  
MaxMsgSize);
```

Lança `MQException`.

Recupera uma mensagem do tópico.

Esse método usa uma instância padrão de `MQGetMessageOptions` para executar a obtenção. A opção de mensagem usada é `MQGMO_NOWAIT`

Se a obtenção falhar, o objeto `MQMessage` permanecerá inalterado. Se ele for bem-sucedido, as partes do descritor de mensagens e de dados da mensagem do `MQMessage` serão substituídas pelo descritor de mensagens e pelos dados da mensagem recebida.

Todas as chamadas para IBM MQ de um determinado `MQQueueManager` são síncronas. Portanto, se você executar um `get` com `wait`, todos os outros encadeamentos usando o mesmo `MQQueueManager` serão bloqueados de fazer chamadas IBM MQ adicionais até que a chamada `Get` seja realizada..

Se você precisar de vários encadeamentos para acessar o IBM MQ simultaneamente, cada encadeamento deverá criar seu próprio objeto `MQQueueManager`

### **mensagem**

Contém o descritor de mensagens e os dados da mensagem retornados. Alguns dos campos no descritor de mensagem são parâmetros de entrada. É importante assegurar que os parâmetros de entrada `MessageId` e `CorrelationId` sejam configurados conforme necessário

Um cliente reconectável retorna o código de razão `MQRC_BACKED_OUT` após a reconexão bem-sucedida para mensagens recebidas em `MQGM_SYNCPOINT`.

### **Opções getMessage**

Opções que controlam a ação de obtenção.

Usar a opção `MQC.MQGMO_CONVERT` pode resultar em uma exceção com o código de razão `MQC.MQRC_CONVERTED_STRING_TOO_BIG` ao converter de códigos de caracteres de byte único para códigos de byte duplo. Neste caso, a mensagem é copiada para o buffer sem conversão

Se *getMessageOptions* não for especificado, a opção de mensagem usada será `MQGMO_NOWAIT`

Se você usar a opção `MQGMO_LOGICAL_ORDER` em um cliente reconectável, o código de razão `MQRC_RECONNECT_INCOMPATIBLE` será retornado

### **MaxMsgTamanho**

A maior mensagem que este objeto de mensagem deve receber. Se a mensagem na fila for maior que este tamanho, uma de duas coisas ocorrerá:

- Se a sinalização `MQGMO_ACCEPT_TRUNCATED_MSG` for configurada no objeto `MQGetMessageOptions`, a mensagem será preenchida com o máximo de dados da mensagem possível. Uma exceção é lançada com o código de conclusão `MQCC_WARNING` e o código de razão `MQRC_TRUNCATED_MSG_ACCEPTED`
- Se o sinalizador `MQGMO_ACCEPT_TRUNCATED_MSG` não for configurado, a mensagem permanecerá na fila. Uma exceção é lançada com o código de conclusão `MQCC_WARNING` e o código de razão `MQRC_TRUNCATED_MSG_FAILED`

Se *MaxMsgSize* não for especificado, a mensagem inteira será recuperada

## Construtores

```
public MQTopic(MQQueueManager queueManager, MQDestination destination, string
topicName, string topicObject, int options);
public MQTopic(MQQueueManager queueManager, MQDestination destination, string
topicName, string topicObject, int options, string alternateUserId);
public MQTopic(MQQueueManager queueManager, MQDestination destination, string
topicName, string topicObject, int options, string alternateUserId, string
subscriptionName);
public MQTopic(MQQueueManager queueManager, MQDestination destination, string
topicName, string topicObject, int options, string alternateUserId, string
subscriptionName, System.Collections.Hashtable properties);
public MQTopic(MQQueueManager queueManager, string topicName, string
topicObject, int openAs, int options);
public MQTopic(MQQueueManager queueManager, string topicName, string
topicObject, int openAs, int options, string alternateUserId);
public MQTopic(MQQueueManager queueManager, string topicName, string
topicObject, int options, string alternateUserId, string subscriptionName);
public MQTopic(MQQueueManager queueManager, string topicName, string
topicObject, int options, string alternateUserId, string subscriptionName,
System.Collections.Hashtable properties);
```

Acesse um tópico em *queueManager*

Os objetos MQTopic estão intimamente relacionados aos objetos do tópico administrativo, que às vezes são chamados de objetos do tópico Na entrada, o topicObject aponta para um objeto do tópico administrativo O construtor MQTopic obtém uma sequência de tópicos do objeto do tópico e a combina com topicName para criar um nome do tópico. topicObject ou topicName pode ser nulo. O nome do tópico é correspondido com a árvore de tópicos e o nome do objeto do tópico administrativo correspondente mais próximo é retornado em topicObject

Os tópicos associados ao objeto MQTopic são o resultado da combinação de duas sequências de tópicos. A primeira sequência de tópicos é definida pelo objeto do tópico administrativo identificado por *topicObject*.. A segunda sequência de tópicos é *topicString* A sequência de tópicos resultante associada com o objeto MQTopic pode identificar vários tópicos incluindo curingas

Dependendo se o tópico é aberto para publicação ou assinatura, é possível usar os métodos MQTopic .Put para publicar em tópicos ou métodos MQTopic .Get para receber publicações em tópicos.. Se você desejar publicar e assinar o mesmo tópico, deverá acessar o tópico duas vezes, uma vez para publicação e uma vez para assinatura.

Se você criar um objeto MQTopic para assinatura, sem fornecer um objeto MQDestination , uma assinatura gerenciada será assumida. Se você passar uma fila como um objeto MQDestination , uma assinatura não gerenciada será assumida Deve-se assegurar que as opções de assinatura configuradas sejam consistentes com a assinatura que está sendo gerenciada ou não gerenciada.

### **queueManager**

O gerenciador de filas no qual acessar um tópico

### **destino**

*destination* é uma instância MQQueue . Ao fornecer *destination*, MQTopic é aberto como uma assinatura não gerenciada. As publicações sobre o tópico são entregues à fila acessada como *destination*.

### **topicName**

Uma sequência de tópicos que é a segunda parte do nome do tópico.. O *topicName* é concatenado com a sequência de tópicos definida no objeto do tópico administrativo *topicObject* É possível configurar *topicName* como nulo, em cujo caso o nome do tópico é definido pela sequência de tópicos em *topicObject*.

### **topicObject**

Na entrada, *topicObject* é o nome do objeto do tópico que contém a sequência de tópicos que forma a primeira parte do nome do tópico A sequência de tópicos em *topicObject* é

concatenada com *topicName* As regras para construir sequências de tópicos são definidas em Combinando sequências de tópicos.

Na saída, *topicObject* contém o nome do objeto do tópico administrativo que é a correspondência mais próxima na árvore de tópicos para o tópico identificado pela sequência de tópicos

### **openAs**

Acesse o tópico para publicar ou assinar.. O parâmetro pode conter apenas uma dessas opções:

- MQC.MQTOPIC\_OPEN\_AS\_SUBSCRIPTION
- MQC.MQTOPIC\_OPEN\_AS\_PUBLICATION

### **opções**

Combine as opções que controlam a abertura do tópico para publicação ou assinatura. Use constantes MQC.MQSO\_\* para acessar um tópico para assinatura e constantes MQC.MQOO\_\* para acessar um tópico para publicação.

Se mais de uma opção for necessária, inclua os valores juntos ou combine os valores de opção usando o operador OR bit a bit

### **alternateUserId**

Especifique o ID do usuário alternativo usado para verificar a autorização necessária para concluir a operação. Deve-se especificar *alternateUserId*, se MQC.MQOO\_ALTERNATE\_USER\_AUTHORITY ou MQC.MQSO\_ALTERNATE\_USER\_AUTHORITY estiver configurado no parâmetro de opções

### **subscriptionName**

*subscriptionName* será necessário se as opções MQC.MQSO\_DURABLE ou MQC.MQSO\_ALTER forem fornecidas.. Nos dois casos, MQTopic é implicitamente aberto para assinatura. Uma exceção será lançada se o MQC.MQSO\_DURABLE estiver configurado e a assinatura existir ou se MQC.MQSO\_ALTER estiver configurado e a assinatura não existir.

### **propriedades**

Configure qualquer uma das propriedades de assinatura especiais listadas usando uma tabela de hash Entradas especificadas na tabela hash são atualizadas com valores de saída. Entradas não são incluídas na tabela hash para relatar valores de saída.

- MQC.MQSUB\_PROP\_ALTERNATE\_SECURITY\_ID
- MQC.MQSUB\_PROP\_SUBSCRIPTION\_EXPIRY
- MQC.MQSUB\_PROP\_SUBSCRIPTION\_USER\_DATA
- MQC.MQSUB\_PROP\_SUBSCRIPTION\_CORRELATION\_ID
- MQC.MQSUB\_PROP\_PUBLICATION\_PRIORITY
- MQC.MQSUB\_PROP\_PUBLICATION\_ACCOUNTING\_TOKEN
- MQC.MQSUB\_PROP\_PUBLICATION\_APPLICATIONID\_DATA

```

public MQTopic MQQueueManager.AccessTopic(MQDestination destination, string
topicName, string topicObject, int options);
public MQTopic MQQueueManager.AccessTopic(MQDestination destination, string
topicName, string topicObject, int options, string alternateUserId);
public MQTopic MQQueueManager.AccessTopic(MQDestination destination, string
topicName, string topicObject, int options, string alternateUserId, string
subscriptionName);
public MQTopic MQQueueManager.AccessTopic(MQDestination destination, string
topicName, string topicObject, int options, string alternateUserId, string
subscriptionName, System.Collections.Hashtable properties);
public MQTopic MQQueueManager.AccessTopic(string topicName, string topicObject,
int openAs, int options);
public MQTopic MQQueueManager.AccessTopic(string topicName, string topicObject,
int openAs, int options, string alternateUserId);
public MQTopic MQQueueManager.AccessTopic(string topicName, string topicObject,
int options, string alternateUserId, string subscriptionName);
public MQTopic MQQueueManager.AccessTopic(string topicName, string
topicObject, int options, string alternateUserId, string subscriptionName,
System.Collections.Hashtable properties);

```

Acesse um tópico neste gerenciador de filas

Os objetos MQTopic estão intimamente relacionados aos objetos do tópico administrativo, que às vezes são chamados de objetos do tópico Na entrada, o topicObject aponta para um objeto do tópico administrativo O construtor MQTopic obtém uma sequência de tópicos do objeto do tópico e a combina com topicName para criar um nome do tópico. topicObject ou topicName pode ser nulo. O nome do tópico é correspondido com a árvore de tópicos e o nome do objeto do tópico administrativo correspondente mais próximo é retornado em topicObject

Os tópicos associados ao objeto MQTopic são o resultado da combinação de duas sequências de tópicos. A primeira sequência de tópicos é definida pelo objeto do tópico administrativo identificado por topicObject.. A segunda sequência de tópicos é topicString A sequência de tópicos resultante associada com o objeto MQTopic pode identificar vários tópicos incluindo curingas

Dependendo se o tópico é aberto para publicação ou assinatura, é possível usar os métodos MQTopic .Put para publicar em tópicos ou métodos MQTopic .Get para receber publicações em tópicos.. Se você deseja publicar e assinar o mesmo tópico, deverá acessar o tópico duas vezes, uma vez para publicação e uma vez para assinatura.

Se você criar um objeto MQTopic para assinatura, sem fornecer um objeto MQDestination , uma assinatura gerenciada será assumida. Se você passar uma fila como um objeto MQDestination , uma assinatura não gerenciada será assumida Deve-se assegurar que as opções de assinatura configuradas sejam consistentes com a assinatura que está sendo gerenciada ou não gerenciada.

#### **destino**

*destination* é uma instância MQQueue . Ao fornecer *destination*, MQTopic é aberto como uma assinatura não gerenciada. As publicações sobre o tópico são entregues à fila acessada como *destination*.

#### **topicName**

Uma sequência de tópicos que é a segunda parte do nome do tópico.. O *topicName* é concatenado com a sequência de tópicos definida no objeto do tópico administrativo *topicObject* É possível configurar *topicName* como nulo, em cujo caso o nome do tópico é definido pela sequência de tópicos em *topicObject*.

#### **topicObject**

Na entrada, *topicObject* é o nome do objeto do tópico que contém a sequência de tópicos que forma a primeira parte do nome do tópico A sequência de tópicos em *topicObject* é concatenada com *topicName* As regras para construir sequências de tópicos são definidas em Combinando sequências de tópicos.

Na saída, *topicObject* contém o nome do objeto do tópico administrativo que é a correspondência mais próxima na árvore de tópicos para o tópico identificado pela sequência de tópicos

### openAs

Acesse o tópico para publicar ou assinar.. O parâmetro pode conter apenas uma dessas opções:

- MQC.MQTOPIC\_OPEN\_AS\_SUBSCRIPTION
- MQC.MQTOPIC\_OPEN\_AS\_PUBLICATION

### opções

Combine as opções que controlam a abertura do tópico para publicação ou assinatura. Use constantes MQC.MQSO\_\* para acessar um tópico para assinatura e constantes MQC.MQOO\_\* para acessar um tópico para publicação.

Se mais de uma opção for necessária, inclua os valores juntos ou combine os valores de opção usando o operador OR bit a bit

### alternateUserId

Especifique o ID do usuário alternativo usado para verificar a autorização necessária para concluir a operação. Deve-se especificar *alternateUserId*, se MQC.MQOO\_ALTERNATE\_USER\_AUTHORITY ou MQC.MQSO\_ALTERNATE\_USER\_AUTHORITY estiver configurado no parâmetro de opções

### subscriptionName

*subscriptionName* será necessário se as opções MQC.MQSO\_DURABLE ou MQC.MQSO\_ALTER forem fornecidas.. Nos dois casos, MQTopic é implicitamente aberto para assinatura. Uma exceção será lançada se o MQC.MQSO\_DURABLE estiver configurado e a assinatura existir ou se MQC.MQSO\_ALTER estiver configurado e a assinatura não existir.

### propriedades

Configure qualquer uma das propriedades de assinatura especiais listadas usando uma tabela de hash Entradas especificadas na tabela hash são atualizadas com valores de saída. Entradas não são incluídas na tabela hash para relatar valores de saída.

- MQC.MQSUB\_PROP\_ALTERNATE\_SECURITY\_ID
- MQC.MQSUB\_PROP\_SUBSCRIPTION\_EXPIRY
- MQC.MQSUB\_PROP\_SUBSCRIPTION\_USER\_DATA
- MQC.MQSUB\_PROP\_SUBSCRIPTION\_CORRELATION\_ID
- MQC.MQSUB\_PROP\_PUBLICATION\_PRIORITY
- MQC.MQSUB\_PROP\_PUBLICATION\_ACCOUNTING\_TOKEN
- MQC.MQSUB\_PROP\_PUBLICATION\_APPLICATIONID\_DATA

## interface do IMQObjectTrigger.NET

Implemente IMQObjectTrigger para processar mensagens transmitidas pelo monitor **runmqdmn.NET**.

### Interface

```
public interface IBM.WMQMonitor.IMQObjectTrigger();
```

Dependendo de o controle do ponto de sincronização ser especificado no comando **runmqdmn**, a mensagem será removida da fila antes ou após o método Execute retornar.

## Métodos

**void Execute (MQQueueManager queueManager, MQQueue queue, MQMessage message, string param);**

**queueManager**

O gerenciador de filas que hospeda a fila que está sendo monitorada

**queue**

Fila sendo monitorada.

**mensagem**

Mensagem lida a partir da fila.

**parâmetro**

Dados transmitidos de UserParameter.

## interface do MQC.NET

Consulte uma constante MQI prefixando o nome da constante com MQC.. MQC define todas as constantes usadas pelo MQI..

### Interface

```
System.Object
└─ IBM.WMQ.MQC
```

```
public interface IBM.WMQ.MQC extends System.Object;
```

### exemplo

```
MQQueue queue;  
queue.closeOptions = MQC.MQCO_DELETE;
```

## Identificadores do conjunto de caracteres para aplicativos .NET

Descrições dos conjuntos de caracteres que você pode selecionar para codificar mensagens do .NET IBM MQ

conjunto de caracteres	Descrição
37	ibm037
437	ibm437 / PC Original
500	ibm500
819	iso-8859-1 / latin1 / ibm819
1200	Unicode
1208	UTF-8
273	ibm273
277	ibm277
278	ibm278
280	ibm280
284	ibm284



<b>conjunto de caracteres</b>	<b>Descrição</b>
285	ibm285
297	ibm297
420	ibm420
424	ibm424
737	ibm737 / PC Grego
775	ibm775 / PC Báltico
813	iso-8859-7 / grego / ibm813
838	ibm838
850	ibm850 / PC Latin 1
852	ibm852 / PC Latin 2
855	ibm855 / PC Cirílico
856	ibm856
857	ibm857 / PC Turco
860	ibm860 / PC Português
861	ibm861 / PC Islandês
862	ibm862 / PC Hebraico
863	ibm863 / PC Francês do Canadá
864	ibm864 / PC Árabe
865	ibm865 / PC Nórdico
866	ibm866 / PC Russo
868	ibm868
869	ibm869 / PC Grego Moderno
870	ibm870
871	ibm871
874	ibm874
875	ibm875
912	iso-8859-2 / latin2 / ibm912
913	iso-8859-3 / latin3 / ibm913
914	iso-8859-4 / latin4 / ibm914
915	iso-8859-5 / cirílico / ibm915
916	iso-8859-8 / hebraico / ibm916
918	ibm918
920	iso-8859-9 / latin5 / ibm920
921	ibm921
922	ibm922

<b>conjunto de caracteres</b>	<b>Descrição</b>
930	ibm930
932	PC Japonês
933	ibm933
935	ibm935
937	ibm937
939	ibm939
942	ibm942
943	ibm943
948	ibm948
949	ibm949
950	ibm950 / Big 5 Chinês Tradicional
954	EUCJIS
964	ibm964 / CNS 11643 Chinês Tradicional
970	ibm970
1006	ibm1006
1025	ibm1025
1026	ibm1026
1089	iso-8859-6 / árabe / ibm1089
1097	ibm1097
1098	ibm1098
1112	ibm1112
1122	ibm1122
1123	ibm1123
1124	ibm1124
1250	Windows Latim 2
1251	Windows Cirílico
1252	Windows Latim 1
1253	Windows Grego
1254	Windows Turco
1255	Windows Hebraico
1256	Windows Árabe
1257	Windows Báltico
1258	Windows Vietnamita
1381	ibm1381
1383	ibm1383

conjunto de caracteres	Descrição
2022	JIS
5601	ksc-5601 Coreano
33722	ibm33722

## Classes IBM MQ C++

As classes C++ IBM MQ encapsulam o Message Queue Interface (MQI) IBM MQ . Há um único arquivo de cabeçalho C++, **imqi.hpp**, que abrange todas essas classes.

Para cada classe, as seguintes informações são mostradas:

### Diagrama de hierarquia de classe

Um diagrama de classes mostrando a classe em sua relação de herança com suas classes pai imediatas, se houver.

### Outras classes relevantes

Links de documentos para outras classes relevantes, como classes-pai e as classes de objetos usadas em assinaturas de método

### Atributos do Objeto

Atributos da classe.. Eles são adicionais aos atributos definidos para quaisquer classes-pai Muitos atributos refletem IBM MQ membros da estrutura de dados (consulte “Referência cruzada de C++ e MQI” na página 1848 ). Para obter descrições detalhadas, consulte “atributos de objetos” na página 827..

### Construtores

Assinaturas dos métodos especiais utilizados para criar um objeto da classe.

### Métodos de objeto (público)

Assinaturas de métodos que requerem uma instância da classe para sua operação e que não possuem restrições de uso..

Quando aplicável, as informações a seguir também são mostradas:

### Métodos de classe (público).

Assinaturas de métodos que não requerem uma instância da classe para sua operação, e que não possuem restrições de uso

### Métodos sobrecarregados (classe pai)

Assinaturas desses métodos virtuais que são definidos em classes pai, mas exibem um comportamento diferente, polimórfico, para essa classe

### Métodos de objeto (protegidos)

Assinaturas de métodos que requerem uma instância da classe para sua operação e são reservadas para uso pelas implementações de classes derivadas Esta seção é de interesse apenas para escritores de classe, em oposição aos usuários de classe.

### Dados do objeto (protegidos).

Detalhes de implementação para dados de instância de objeto disponíveis para as implementações de classes derivadas. Esta seção é de interesse apenas para escritores de classe, em oposição aos usuários de classe.

### Códigos de Razão

Valores MQRC\_ \* (consulte [conclusão da API e códigos de razão](#)) que podem ser esperados a partir desses métodos que falham. Para uma lista exaustiva de códigos de razão que podem ocorrer para um objeto de uma classe, consulte a documentação da classe pai. A lista documentada de códigos de razão para uma classe não inclui os códigos de razão para classes pai.

### Nota:

1. Os objetos dessas classes não são thread-safe Isso garante um desempenho ideal, mas tome cuidado para não acessar nenhum objeto de mais de um encadeamento.

2. É recomendado que, para um programa multiencadeado, um objeto `ImqQueueManager` separado seja usado para cada encadeamento. Cada objeto do gerenciador deve ter sua própria coleção independente de outros objetos, assegurando que os objetos em encadeamentos diferentes sejam isolados uns dos outros

As classes são:

- [“Classe C++ do Registro `ImqAuthentication`”](#) na página 1865
- [“Classe C++ `ImqBinary`”](#) na página 1867
- [“Classe C++ `ImqCache`”](#) na página 1869
- [“Classe C++ `ImqChannel`”](#) na página 1872
- [“Classe C++ `ImqCICSBridgeHeader`”](#) na página 1878
- [“`ImqDeadLetterHeader` Classe C++”](#) na página 1884
- [“`ImqDistributionList` classe C++”](#) na página 1887
- [“Classe C++ `ImqError`”](#) na página 1888
- [“Classe C++ `ImqGetMessageOptions`”](#) na página 1889
- [“Classe C++ `ImqHeader`”](#) na página 1893
- [“Classe C++ `ImqIMSBridgeHeader`”](#) na página 1894
- [“Classe C++ `ImqItem`”](#) na página 1897
- [“Classe C++ `ImqMessage`”](#) na página 1899
- [“Classe C++ do Rastreador `ImqMessage`”](#) na página 1906
- [“Classe C++ `ImqNamelist`”](#) na página 1909
- [“Classe C++ `ImqObject`”](#) na página 1910
- [“Classe C++ `ImqProcess`”](#) na página 1916
- [“Classe C++ `ImqPutMessageOptions`”](#) na página 1918
- [“Classe C++ `ImqQueue`”](#) na página 1920
- [“Classe C++ do Gerenciador `ImqQueue`”](#) na página 1931
- [“Classe C++ do cabeçalho `ImqReference`”](#) na página 1947
- [“Classe C++ `ImqString`”](#) na página 1950
- [“Classe C++ `ImqTrigger`”](#) na página 1956
- [“Classe C++ do cabeçalho `ImqWork`”](#) na página 1958

## Referência cruzada de C++ e MQI

Esta coleção de tópicos contém informações relacionadas C++ ao MQI.

Leia essas informações junto com o [“Tipos de dados usados no MQI”](#) na página 237

Esta tabela relaciona estruturas de dados MQI às classes C++ e arquivos include. Os seguintes tópicos mostram informações de referência cruzada para cada classe C + +. Essas referências cruzadas estão relacionadas ao uso das interfaces processuais subjacentes do IBM MQ . As classes `ImqBinary`, `ImqDistributionList` e `ImqString` não têm atributos que se encaixam nessa categoria e são excluídos

<i>Tabela 845. Estrutura de dados, classe e referência cruzada de arquivo include</i>		
<b>Estrutura de dados</b>	<b>Parâmetro class</b>	<b>Arquivo de inclusão</b>
MQAIR	Registro <code>ImqAuthentication</code>	<code>imqair.hpp</code>
	<code>ImqBinary</code>	<code>imqbin.hpp</code>
	<code>ImqCache</code>	<code>imqcac.hpp</code>
MQCD	<code>ImqChannel</code>	<code>imqchl.hpp</code>

Tabela 845. Estrutura de dados, classe e referência cruzada de arquivo include (continuação)

<b>Estrutura de dados</b>	<b>Parâmetro class</b>	<b>Arquivo de inclusão</b>
MQCIH	ImqCICSBridgeHeader	imqcih.hpp
MQDLH	ImqDeadLetterHeader	imqdlh.hpp
MQOR	Lista de ImqDistribution	imqdst.hpp
	ImqError	imqerr.hpp
MQGMO	ImqGetMessageOptions	imqgmo.hpp
	ImqHeader	imqhdr.hpp
MQIIH	ImqIMSBridgeHeader	imqiih.hpp
	ImqItem	imqitm.hpp
MQMD	ImqMessage	imqmsg.hpp
	Rastreador ImqMessage	imqmtr.hpp
	ImqNamelist	imqnml.hpp
MQOD, MQRR	ImqObject	imqobj.hpp
MQPMO, MQPMR, MQRR	ImqPutMessageOptions	imqpmo.hpp
	ImqProcess	imqpro.hpp
	ImqQueue	imqque.hpp
MQBO, MQCNO, MQCSP	ImqQueueGerenciador	imqmgr.hpp
MQRMH	Cabeçalho ImqReference	imqrfh.hpp
	ImqString	imqstr.hpp
MQTM	ImqTrigger	imqtrg.hpp
MQTMC		
MQTMC2	ImqTrigger	imqtrg.hpp
MQXQH		
MQWIH	Cabeçalho ImqWork	imqwih.hpp

## Referência cruzada do Registro ImqAuthentication

Referência cruzada de atributos, estruturas de dados, campos e chamadas para a classe C++ de registro ImqAuthentication.

Tabela 846. Atributos, estruturas de dados, campos e chamadas

<b>Atributo</b>	<b>Estrutura de dados</b>	<b>Campo</b>	<b>Ligue para</b>
nome de conexão	MQAIR	AuthInfoConnName	MQCONNX
senha	MQAIR	LDAPPassword	MQCONNX
Tipo	MQAIR	AuthInfoType	MQCONNX
nome do usuário	MQAIR	LDAPUserNamePtr	MQCONNX
	MQAIR	Deslocamento de LDAPUserName	MQCONNX

Tabela 846. Atributos, estruturas de dados, campos e chamadas (continuação)

Atributo	Estrutura de dados	Campo	Ligue para
	MQAIR	Comprimento de LDAPUserName	MQCONN

## Referência cruzada de ImqCache

Referência cruzada de atributos e chamadas para a classe C++ ImqCache .

Tabela 847. Atributos e chamadas

Atributo	Ligue para
buffer automático	MQGET
Comprimento do buffer	MQGET
ponteiro do buffer	MQGET, MQPUT
Comprimento dos Dados	MQGET
Deslocamento de dados	MQGET
ponteiro de dados	MQGET
comprimento da mensagem	MQGET, MQPUT

## Referência cruzada ImqChannel

Referência cruzada de atributos, estruturas de dados, campos e chamadas para a classe C++ ImqChannel .

Tabela 848. Atributos, estruturas de dados, campos e chamadas

Atributo	Estrutura de dados	Campo	Ligue para
batimento cardíaco em lote	MQCD	BatchHeartbeat	MQCONN
nome do canal	MQCD	ChannelName	MQCONN
nome de conexão	MQCD	ConnectionName	MQCONN
	MQCD	ShortConnectionNome	MQCONN
Compactação do cabeçalho	MQCD	HdrCompList	MQCONN
intervalo de batimento cardíaco	MQCD	HeartbeatInterval	MQCONN
Intervalo keep-alive	MQCD	KeepAliveInterval	MQCONN
endereço local	MQCD	LocalAddress	MQCONN
comprimento máximo da mensagem	MQCD	MaxMsgLength	MQCONN
Compressão da mensagem	MQCD	MsgCompList	MQCONN
nome de modo	MQCD	ModeName	MQCONN
senha	MQCD	Senha	MQCONN
contagem de saída de recebimento	MQCD		MQCONN
nomes de saída de recebimento	MQCD	ReceiveExit	MQCONN

*Tabela 848. Atributos, estruturas de dados, campos e chamadas (continuação)*

<b>Atributo</b>	<b>Estrutura de dados</b>	<b>Campo</b>	<b>Ligue para</b>
	MQCD	ReceiveExitsDefinido	MQCONN
	MQCD	ReceiveExitPtr	MQCONN
receber dados do usuário	MQCD	ReceiveUserData	MQCONN
	MQCD	ReceiveUserDataPtr	MQCONN
Nome da saída de segurança	MQCD	SecurityExit	MQCONN
dados do usuário de segurança	MQCD	SecurityUserData	MQCONN
contagem de saída de envio	MQCD		MQCONN
enviar nomes de saída	MQCD	SendExit	MQCONN
	MQCD	SendExitsDefinido	MQCONN
	MQCD	SendExitPtr	MQCONN
enviar dados do usuário	MQCD	SendUserData	MQCONN
	MQCD	SendUserDataPtr	MQCONN
SSL CipherSpec	MQCD	sslCipherEspecificação	MQCONN
tipo de autenticação de cliente SSL	MQCD	Autenticação sslClient	MQCONN
Nome do peer SSL	MQCD	sslPeerName	MQCONN
nome do programa de transação	MQCD	TpName	MQCONN
tipo de transporte	MQCD	TransportType	MQCONN
ID do usuário	MQCD	UserIdentifier	MQCONN

## Referência cruzada do ImqCICSBridgeHeader

Referência cruzada de atributos, estruturas de dados e campos para a classe C++ ImqCICSBridgeHeader .

*Tabela 849. Mapeamento de atributos, estruturas de dados e campos*

<b>Atributo</b>	<b>Estrutura de dados</b>	<b>Campo</b>
código de encerramento anormal de ponte	MQCIH	AbendCode
Descritor ADS	MQCIH	AdsDescriptor
identificador de atenção	MQCIH	AttentionId
autenticador	MQCIH	Autenticador
código de conclusão da ponte	MQCIH	Código BridgeCompletion
compensação de erro de ponte	MQCIH	ErrorOffset
código de razão da ponte	MQCIH	BridgeReason
código de cancelamento da ponte,	MQCIH	CancelCode
tarefa conversacional	MQCIH	ConversationalTask
posição do cursor	MQCIH	CursorPosition

Tabela 849. Mapeamento de atributos, estruturas de dados e campos (continuação)

Atributo	Estrutura de dados	Campo
token do recurso	MQCIH	Instalação
instalação manter tempo	MQCIH	FacilityKeepTime
instalações como	MQCIH	FacilityLike
função	MQCIH	Função
obter intervalo de espera	MQCIH	GetWaitInterval
tipo de link	MQCIH	LinkType
próximo identificador de transação	MQCIH	NextTransactionId
comprimento dos dados de saída	MQCIH	OutputDataLength
formato de resposta para	MQCIH	ReplyToFormat
código de retorno da ponte	MQCIH	ReturnCode
código de início	MQCIH	StartCode
status de encerramento da tarefa	MQCIH	TaskEndStatus
identificador de transação	MQCIH	TransactionId
controle de uow	MQCIH	UowControl
versão	MQCIH	Versão

## Referência cruzada de ImqDeadLetterHeader

Referência cruzada de atributos, estruturas de dados e campos para a classe C++ ImqDeadLetterHeader .

Tabela 850. Mapeamento de atributos, estruturas de dados e campos

Atributo	Estrutura de dados	Campo
código de razão de devoluções	MQDLH	Razão
Nome do gerenciador de filas de destino	MQDLH	DestQMgrName
Nome da Fila de Destino	MQDLH	DestQName
Colocar nome do aplicativo	MQDLH	PutApplName
Colocar tipo de aplicativo	MQDLH	PutApplType
Data de colocação	MQDLH	PutDate
Horário de colocação	MQDLH	PutTime

## Referência cruzada ImqError

Referência cruzada de atributos e chamadas para a classe C++ ImqError .

Tabela 851. Atributos e chamadas

Atributo	Ligue para
código de conclusão	MQBACK, MQBEGIN, MQCLOSE, MQCMIT, MQCONN, MQCONNX, MQDISC, MQGET, MQINQ, MQOPEN, MQPUT, MQSET
código de razão	MQBACK, MQBEGIN, MQCLOSE, MQCMIT, MQCONN, MQCONNX, MQDISC, MQGET, MQINQ, MQOPEN, MQPUT, MQSET



## Referência cruzada ImqGetMessageOptions

Referência cruzada de atributos, estruturas de dados e campos para a classe C++ ImqGetMessageOptions .

*Tabela 852. Mapeamento de atributos, estruturas de dados e campos*

Atributo	Estrutura de dados	Campo
status do grupo	MQGMO	GroupStatus
opções de correspondência	MQGMO	MatchOptions
message token (token de mensagem)	MQGMO	MessageToken
opções	MQGMO	Opções
Nome da fila resolvida	MQGMO	ResolvedQName
Comprimento retornado	MQGMO	ReturnedLength
segmentação	MQGMO	Segmentação
status do segmento	MQGMO	SegmentStatus
	MQGMO	Signal1
	MQGMO	Signal2
participação de ponto de sincronização	MQGMO	Opções
intervalo de espera	MQGMO	WaitInterval

## Referência cruzada ImqHeader

Referência cruzada de atributos, estruturas de dados e campos para a classe C++ ImqHeader .

*Tabela 853. Mapeamento de atributos, estruturas de dados e campos*

Atributo	Estrutura de dados	Campo
conjunto de caracteres	MQDLH, MQIIH	CodedCharSetId
codificação	MQDLH, MQIIH	Codificação
format	MQDLH, MQIIH	Formato
sinalizações de cabeçalho	MQIIH, MQRMH	Sinalizadores

## Referência cruzada do ImqIMSBridgeHeader

Referência cruzada de atributos, estruturas de dados e campos para a classe C++ do Registro ImqAuthentication.

*Tabela 854. Mapeamento de atributos, estruturas de dados e campos*

Atributo	Estrutura de dados	Campo
autenticador	MQIIH	Autenticador
Modo Comprometer	MQIIH	CommitMode
Intervenção terminal lógica	MQIIH	LTermOverride
Nome de mapa dos Serviços de Formato de Mensagens	MQIIH	MFSTMapName
formato de resposta para	MQIIH	ReplyToFormat

<i>Tabela 854. Mapeamento de atributos, estruturas de dados e campos (continuação)</i>		
<b>Atributo</b>	<b>Estrutura de dados</b>	<b>Campo</b>
escopo de segurança	MQIIH	SecurityScope
id da instância de transação	MQIIH	TranInstanceId
ESTADO DA TRANSAÇÃO	MQIIH	TranState

### **Referência cruzada ImqItem**

Referência cruzada de atributos e chamadas para a classe C++ ImqItem .

<i>Tabela 855. Atributos e chamadas</i>	
<b>Atributo</b>	<b>Ligue para</b>
id da estrutura	MQGET

### **Referência cruzada ImqMessage**

Referência cruzada de atributos, estruturas de dados, campos e chamadas para a classe C++ ImqMessage .

<i>Tabela 856. Atributos, estruturas de dados, campos e chamadas</i>			
<b>Atributo</b>	<b>Estrutura de dados</b>	<b>Campo</b>	<b>Ligue para</b>
Dados do ID do aplicativo	MQMD	ApplIdentityData	
Dados de origem do aplicativo	MQMD	ApplOriginData	
contagem de restauração	MQMD	BackoutCount	
conjunto de caracteres	MQMD	CodedCharSetId	
codificação	MQMD	Codificação	
expiração	MQMD	Expiração	
format	MQMD	Formato	
Sinalizações de mensagem	MQMD	MsgFlags	
tipo de mensagem	MQMD	MsgType	
deslocamento	MQMD	Offset	
Comprimento original	MQMD	OriginalLength	
persistence	MQMD	Persistence	
priority	MQMD	Priority	
Colocar nome do aplicativo	MQMD	PutApplName	
Colocar tipo de aplicativo	MQMD	PutApplType	
Data de colocação	MQMD	PutDate	
Horário de colocação	MQMD	PutTime	
Nome do Gerenciador de Filas de Resposta	MQMD	ReplyToQMgr	
Nome da Fila de Resposta	MQMD	ReplyToQ	
relatório	MQMD	Relatório	

<i>Tabela 856. Atributos, estruturas de dados, campos e chamadas (continuação)</i>			
<b>Atributo</b>	<b>Estrutura de dados</b>	<b>Campo</b>	<b>Ligue para</b>
número de sequência	MQMD	MsgSeqNumber	
comprimento total da mensagem		DataLength	MQGET
ID do usuário	MQMD	UserIdentifier	

### **Referência cruzada do ImqMessageTracker**

Referência cruzada de atributos, estruturas de dados e campos para a classe C++ do ImqMessageTracker.

<i>Tabela 857. Mapeamento de atributos, estruturas de dados e campos</i>		
<b>Atributo</b>	<b>Estrutura de dados</b>	<b>Campo</b>
token de conta	MQMD	AccountingToken
ID de correlação	MQMD	CorrelId
feedback	MQMD	Feedback
ID de grupo	MQMD	GroupId
ID da mensagem	MQMD	MsgId

### **Referência cruzada ImqNamelist**

Referência cruzada de atributos, consultas e chamadas para a classe C++ ImqNamelist

<i>Tabela 858. Atributos, consultas e chamadas</i>		
<b>Atributo</b>	<b>Consultar</b>	<b>Ligue para</b>
Contagem de nomes	MQIA_NAME_COUNT	MQINQ
Nome da lista de nomes	MQCA_NAMELIST_NAME	MQINQ

### **Referência cruzada ImqObject**

Referência cruzada de atributos, estruturas de dados, campos, consultas e chamadas para a classe C++ ImqObject .

<i>Tabela 859. Atributos, estruturas de dados, campos, consultas e chamadas</i>				
<b>Atributo</b>	<b>Estrutura de dados</b>	<b>Campo</b>	<b>Consultar</b>	<b>Ligue para</b>
Data da alteração			MQCA_ALTERATION_DATE	MQINQ
Hora da alteração			MQCA_ALTERATION_TIME	MQINQ
ID do usuário alternativo	MQOD	AlternateUserId		
ID de segurança alternativo				
opções de fechamento				MQCLOSE
descrição			MQCA_Q_DESC, MQCA_Q_MGR_DESC, MQCA_PROCESS_DESC	MQINQ

<i>Tabela 859. Atributos, estruturas de dados, campos, consultas e chamadas (continuação)</i>				
<b>Atributo</b>	<b>Estrutura de dados</b>	<b>Campo</b>	<b>Consultar</b>	<b>Ligue para</b>
nome	MQOD	ObjectName	MQCA_Q_MGR_NAME, MQCQ_Q_NAME, MQCA_PROCESS_NAME	MQINQ
Opções de abertura				MQOPEN
status de abertura				MQOPEN, MQCLOSE
Identificador do Gerenciador de Filas	Identificador do Gerenciador de Filas		MQCA_Q_MGR_IDENTIFIER	MQINQ

### **Referência cruzada ImqProcess**

Referência cruzada de atributos, consultas e chamadas para a classe C++ do registro ImqAuthentication.

<i>Tabela 860. Atributos, consultas e chamadas</i>		
<b>Atributo</b>	<b>Consultar</b>	<b>Ligue para</b>
ID do aplicativo	MQCA_APPL_ID	MQINQ
tipo de aplicativo	MQIA_APPL_TYPE	MQINQ
Dados do ambiente	MQCA_ENV_DATA	MQINQ
dados do usuário	MQCA_USER_DATA	MQINQ

### **Referência cruzada de ImqPutMessageOptions .**

Referência cruzada de atributos, estruturas de dados e campos para a classe C++ do Registro ImqAuthentication.

<i>Tabela 861. Mapeamento de atributos, estruturas de dados e campos</i>		
<b>Atributo</b>	<b>Estrutura de dados</b>	<b>Campo</b>
referência de contexto	MQPMO	Context
	MQPMO	InvalidDestCount
	MQPMO	KnownDestCount
opções	MQPMO	Opções
campos de registro	MQPMO	PutMsgRecFields
Nome do Gerenciador de Filas Resolvido	MQPMO	ResolvedQMgrName
Nome da fila resolvida	MQPMO	ResolvedQName
	MQPMO	Tempo Limite
	MQPMO	UnknownDestCount
participação de ponto de sincronização	MQPMO	Opções

## Referência cruzada ImqQueue

Referência cruzada de atributos, estruturas de dados, campos, consultas e chamadas para a classe C++ ImqQueue .

*Tabela 862. Referência cruzada ImqQueue*

Atributo	Estrutura de dados	Campo	Consultar	Ligue para
Nome do reenfileiramento de restauração			MQCA_BACKOUT_REQ_Q_NAME	MQINQ
limite de restauração			MQIA_BACKOUT_THRESHOLD	MQINQ
Nome da Fila Base			MQCA_BASE_Q_NAME	MQINQ
nome do cluster			MQCA_CLUSTER_NAME	MQINQ
Nome da lista de nomes do cluster			MQCA_CLUSTER_NAMELIST	MQINQ
Classificação da Carga de Trabalho de Cluster			MQIA_CLWL_Q_RANK	MQINQ
Prioridade da Carga de Trabalho de Cluster			MQIA_CLWL_Q_PRIORITY	MQINQ
Fila de uso de carga de trabalho do cluster			MQIA_CLWL_USEQ	MQINQ
data de criação			MQCA_CREATION_DATE	MQINQ
horário de criação			MQCA_CREATION_TIME	MQINQ
Profundidade atual			MQIA_CURRENT_Q_DEPTH	MQINQ
Ligação Padrão			MQIA_DEF_BIND	MQINQ
Opções de abertura de entrada padrão			MQIA_DEF_INPUT_OPEN_OPTION	MQINQ
Persistência padrão			MQIA_DEF_PERSISTENCE	MQINQ
Prioridade padrão			MQIA_DEF_PRIORITY	MQINQ
tipo de definição			MQIA_DEFINITION_TYPE	MQINQ
evento de alta profundidade			MQIA_Q_DEPTH_HIGH_EVENT	MQINQ
Limite superior de profundidade			MQIA_Q_DEPTH_HIGH_LIMIT	MQINQ
evento de baixa profundidade			MQIA_Q_DEPTH_LOW_EVENT	MQINQ
Limite inferior de profundidade			MQIA_Q_DEPTH_LOW_LIMIT	MQINQ
evento máximo de profundidade			MQIA_Q_DEPTH_MAX_LIMIT	MQINQ

Tabela 862. Referência cruzada ImqQueue (continuação)

Atributo	Estrutura de dados	Campo	Consultar	Ligue para
listas de distribuição			MQIA_DIST_LISTS	MQINQ, MQSET
Nome da fila dinâmica	MQOD	DynamicQName		
Forçar obtenção de recuperação			MQIA_HARDEN_GET_BACKOUT	MQINQ
Tipo de índice			MQIA_INDEX_TYPE	MQINQ
inibe obter			MQIA_INIBBIT_GET	MQINQ, MQSET
inibir colocação			MQIA_INIBBIT_PUT	MQINQ, MQSET
Nome da Fila de Inicialização			MQCA_INITIATION_Q_NAME	MQINQ
Profundidade máxima			MQIA_MAX_Q_DEPTH	MQINQ
comprimento máximo da mensagem			MQIA_MAX_MSG_LENGTH	MQINQ
Seqüência de entrega de mensagens			MQIA_MSG_DELIVERY_SEQUENCE	MQINQ
próxima fila distribuída				
Nenhuma Classe de Mensagem Persistente			MQIA_NPM_CLASS	MQINQ
Contagem de entradas abertas			MQIA_OPEN_INPUT_COUNT	MQINQ
Contagem de saídas abertas			MQIA_OPEN_OUTPUT_COUNT	MQINQ
fila distribuída anterior				
nome do processo			MQCA_PROCESS_NAME	MQINQ
Enfileiramento de Contas			MQIA_ACCOUNTING_Q	MQINQ
Nome do Gerenciador de Fila	MQOD	ObjectQMgrName		
Monitoramento de Fila			MQIA_MONITORING_Q	MQINQ
estatísticas de fila			MQIA_STATISTICS_Q	MQINQ
tipo de fila			MQIA_Q_TYPE	MQINQ
Nome do Gerenciador de Filas Remotas			MQCA_REMOTE_Q_MGR_NAME	MQINQ

<i>Tabela 862. Referência cruzada ImqQueue (continuação)</i>				
<b>Atributo</b>	<b>Estrutura de dados</b>	<b>Campo</b>	<b>Consultar</b>	<b>Ligue para</b>
Nome da fila remota			MQCA_REMOTE_Q_NAME	MQINQ
Nome do Gerenciador de Filas Resolvido	MQOD	ResolvedQMgrName		
Nome da fila resolvida	MQOD	ResolvedQName		
intervalo de retenção			MQIA_RETENTION_INTERVAL	MQINQ
escopo			MQIA_SCOPE	MQINQ
intervalo de serviço			MQIA_Q_SERVICE_INTERVAL	MQINQ
evento de intervalo de serviço			MQIA_Q_SERVICE_INTERVAL_EVENT	MQINQ
Capacidade de compartilhamento			MQIA_SHAREABILITY	MQINQ
classe de armazenamento			MQCA_STORAGE_CLASS	MQINQ
Nome da fila de transmissão			MQCA_XMIT_Q_NAME	MQINQ
Controle de ativação			MQIA_TRIGGER_CONTROL	MQINQ, MQSET
dados do acionador			MQCA_TRIGGER_DATA	MQINQ, MQSET
Profundidade de ativação			MQIA_TRIGGER_DEPTH	MQINQ, MQSET
Prioridade de mensagem de ativação			MQIA_TRIGGER_MSG_PRIORITY	MQINQ, MQSET
tipo de acionador			MQIA_TRIGGER_TYPE	MQINQ, MQSET
uso			MQIA_USAGE	MQINQ

## Referência cruzada do Gerenciador ImqQueue

Referência cruzada de atributos, estruturas de dados, campos, consultas e chamadas para a classe C++ do Gerenciador de ImqQueue

<i>Tabela 863. Atributos, estruturas de dados, campos, consultas e chamadas</i>				
<b>Atributo</b>	<b>Estrutura de dados</b>	<b>Campo</b>	<b>Consultar</b>	<b>Ligue para</b>
substituição de conexões de contabilidade			MQIA_ACCOUNTING_CONN_OVERRIDE	MQINQ
intervalo contábil			MQIA_ACCOUNTING_INTERVAL	MQINQ

Tabela 863. Atributos, estruturas de dados, campos, consultas e chamadas (continuação)

<b>Atributo</b>	<b>Estrutura de dados</b>	<b>Campo</b>	<b>Consultar</b>	<b>Ligue para</b>
gravação de atividade			MQIA_ACTIVITY_RECORDING	MQINQ
Adotar nova verificação de MCA			MQIA_ADOPTNEWMCA_CHECK	MQINQ
Adotar novo tipo de MCA			MQIA_ADOPTNEWMCA_TYPE	MQINQ
tipo de autenticação	MQCSP	AuthenticationType		MQCONN
Evento de Autoridade			MQIA_AUTHORITY_EVENT	MQINQ
Iniciar Opções	MQBO	Opções		MQBEGIN
evento de ponte			MQIA_BRIDGE_EVENT	MQINQ
Definição automática de canal			MQIA_CHANNEL_AUTO_DEF	MQINQ
evento de definição automática do canal			MQIA_CHANNEL_AUTO_EVENT	MQIA
Saída de auto-definição de canal			MQIA_CHANNEL_AUTO_EXIT	MQIA
evento do canal			MQIA_CHANNEL_EVENT	MQINQ
Adaptadores do iniciador de canais			MQIA_CHINIT_ADAPTERS	MQINQ
Controle de inicializador de canais			MQIA_CHINIT_CONTROL	MQINQ
Dispatchers do iniciador de canais			MQIA_CHINIT_DISPATCHERS	MQINQ
Início automático de rastreamento do iniciador de canais			MQIA_CHINIT_TRACE_AUTO_START	MQINQ
Tamanho da tabela de rastreamento do iniciador de canais			MQIA_CHINIT_TRACE_TABLE_SIZE	MQINQ
Monitoramento de canal			MQIA_MONITORING_CHANNEL	MQINQ
referência de canal	MQCD	ChannelType		MQCONN
estatísticas do canal			MQIA_STATISTICS_CHANNEL	MQINQ



Tabela 863. Atributos, estruturas de dados, campos, consultas e chamadas (continuação)

Atributo	Estrutura de dados	Campo	Consultar	Ligue para
conjunto de caracteres			MQIA_CODED_CHAR_SET_ID	MQINQ
Monitoramento do Emissor de Cluster			MQIA_MONITORING_AUTO_CLUSSDR	MQINQ
Estatísticas do Emissor de Cluster			MQIA_STATISTICS_AUTO_CLUSSDR	MQINQ
Dados de carga de trabalho do cluster			MQCA_CLUSTER_WORKLOAD_DATA	MQINQ
saída da carga de trabalho do cluster			MQCA_CLUSTER_WORKLOAD_EXIT	MQINQ
Comprimento de carga de trabalho do cluster			MQIA_CLUSTER_WORKLOAD_LENGTH	MQINQ
mru de carga de trabalho do cluster			MQIA_CLWL_MRU_CHANNELS	MQINQ
Fila de uso de carga de trabalho do cluster			MQIA_CLWL_USEQ	MQINQ
evento de comando			MQIA_COMMAND_EVENT	MQINQ
Nome da fila de entrada do comando			MQCA_COMMAND_INPUT_Q_NAME	MQINQ
nível de comando			MQIA_COMMAND_LEVEL	MQINQ
Controle de servidor de comandos			MQIA_CMD_SERVER_CONTROL	MQINQ
Opções de Conexão	MQCNO	Opções		MQCONN, MQCONNX
ID de conexão	MQCNO	ConnectionId		MQCONNX
status da conexão				MQCONN, MQCONNX, MQDISC
tag de conexão	MQCD	ConnTag		MQCONNX
hardware de criptografia	MQSCO	CryptoHardware		MQCONNX
nome da fila de devoluções			MQCA_DEAD_LETTER_Q_NAME	MQINQ
Nome da Fila de Transmissão Padrão			MQCA_DEF_XMIT_Q_NAME	MQINQ

Tabela 863. Atributos, estruturas de dados, campos, consultas e chamadas (continuação)

Atributo	Estrutura de dados	Campo	Consultar	Ligue para
listas de distribuição			MQIA_DIST_LISTS	MQINQ
grupo de DNS			MQCA_DNS_GROUP	MQINQ
dns wlm			MQIA_DNS_WLM	MQINQ
primeiro registro de autenticação	MQSCO	AuthInfoRecOffset		MQCONN
	MQSCO	AuthInfoRecPtr		MQCONN
Evento de Inibição			MQIA_INIBBIT_EVENT	MQINQ
Versão do endereço IP			MQIA_IP_ADDRESS_VERSION	MQINQ
repositório de chaves	MQSCO	KeyRepository		MQCONN
contagem de reconfiguração de chave	MQSCO	KeyResetCount		MQCONN
Cronômetro do listener			MQIA_LISTENER_TIMER	MQINQ
Evento Local			MQIA_LOCAL_EVENT	MQINQ
evento do criador de logs			MQIA_LOGGER_EVENT	MQINQ
Nome do grupo de LUs			MQCA_LU_GROUP_NAME	MQINQ
Nome de LU			MQCA_LU_NAME	MQINQ
lu62 sufixo do braço			MQCA_LU62_ARM_SUFFIX	MQINQ
lu62 canais			MQIA_LU62_CHANNELS	MQINQ
máximo de canais ativos			MQIA_ACTIVE_CHANNELS	MQINQ
Máximo de canais			MQIA_MAX_CHANNELS	MQINQ
Identificadores Máximos			MQIA_MAX_MANIPULADORES	MQINQ
comprimento máximo da mensagem			MQIA_MAX_MSG_LENGTH	MQINQ
Prioridade máxima			MQIA_MAX_PRIORITY	MQINQ
Máx. de mens. não confirm.			MQIA_MAX_UNCOMMITTED_MSGS	MQINQ
Contabilidade do MQI			MQIA_ACCOUNTING_MQI	MQINQ

Tabela 863. Atributos, estruturas de dados, campos, consultas e chamadas (continuação)

Atributo	Estrutura de dados	Campo	Consultar	Ligue para
Estatística de MQI			MQIA_STATISTICS_MQI	MQINQ
máximo da porta de saída			MQIA_OUTBOUND_PORT_MAX	MQINQ
mínimo da porta de saída			MQIA_OUTBOUND_PORT_MIN	MQINQ
senha	MQCSP	CSPPasswordPtr		MQCONNX
	MQCSP	CSPPasswordOffset		MQCONNX
	MQCSP	CSPPasswordLength		MQCONNX
evento de desempenho			MQIA_PERFORMANCE_EVENT	MQINQ
platform			MQIA_PLATFORM	MQINQ
Enfileiramento de Contas			MQIA_ACCOUNTING_Q	MQINQ
Monitoramento de Fila			MQIA_MONITORING_Q	MQINQ
estatísticas de fila			MQIA_STATISTICS_Q	MQINQ
Tempo limite de recebimento			MQIA_RECEIVE_TIMEOUT	MQINQ
mínimo de tempo limite de recebimento			MQIA_RECEIVE_TIMEOUT_MIN	MQINQ
Tipo de tempo limite de recebimento			MQIA_RECEIVE_TIMEOUT_TYPE	MQINQ
Evento Remoto			MQIA_REMOTE_EVENT	MQINQ
nome de repositório			MQCA_REPOSITORY_NAME	MQINQ
Lista de Nomes do Repositório			MQCA_REPOSITORY_NAMELIST	MQINQ
nome do gerenciador de filas compartilhadas			MQIA_SHARED_Q_Q_MGR_NAME	MQINQ
evento ssl			MQIA_SSL_EVENT	MQINQ
ssl fips			MQIA_SSL_FIPS_REQUIRED	MQINQ
Contagem de reconfigurações de chave SSL			MQIA_SSL_RESET_COUNT	MQINQ

Tabela 863. Atributos, estruturas de dados, campos, consultas e chamadas (continuação)

Atributo	Estrutura de dados	Campo	Consultar	Ligue para
evento de parada inicial			MQIA_START_STOP_EVENT	MQINQ
intervalo de estatísticas			MQIA_STATISTICS_INTERVAL	MQINQ
Disponibilidade do ponto de sincronização			MQIA_SYNCPOINT	MQINQ
canais tcp			MQIA_TCP_CHANNELS	MQINQ
Manter TCP ativo			MQIA_TCP_KEEP_ALIVE	MQINQ
Nome do TCP			MQCA_TCP_NAME	MQINQ
Tipo de pilha TCP			MQIA_TCP_STACK_TYPE	MQINQ
Registro de Rota de Rastreo			MQIA_TRACE_ROUTE_RECORDING	MQINQ
Intervalo de ativação			MQIA_TRIGGER_INTERVAL	MQINQ
ID do usuário	MQCSP	CSPUserIdPtr		MQCONN
	MQCSP	Deslocamento de CSPUserId		MQCONN
	MQCSP	Comprimento de CSPUserId		MQCONN

## Referência cruzada do cabeçalho ImqReference

Referência cruzada de atributos, estruturas de dados e campos para a classe C++ do Registro ImqAuthentication.

Tabela 864. Mapeamento de atributos, estruturas de dados e campos

Atributo	Estrutura de dados	Campo
ambiente de destino	MQRMH	DestEnvComprimento, DestEnvDeslocamento
nome do destino	MQRMH	DestNameComprimento, DestNameDeslocamento
ID da instância	MQRMH	ObjectInstanceId
comprimento lógico	MQRMH	DataLogicalLength
deslocamento lógico	MQRMH	DataLogicalOffset
deslocamento lógico 2	MQRMH	DataLogicalOffset2
tipo de referência	MQRMH	ObjectType
ambiente de origem	MQRMH	SrcEnvComprimento, SrcEnvDeslocamento
nome de origem	MQRMH	SrcNameLength, SrcNameOffset

## Referência cruzada ImqTrigger

Referência cruzada de atributos, estruturas de dados e campos para a classe C++ do Registro ImqAuthentication.

*Tabela 865. Mapeamento de atributos, estruturas de dados e campos*

Atributo	Estrutura de dados	Campo
ID do aplicativo	MQTM	ApplId
tipo de aplicativo	MQTM	ApplType
Dados do ambiente	MQTM	EnvData
nome do processo	MQTM	ProcessName
nome da fila	MQTM	QName
dados do acionador	MQTM	TriggerData
dados do usuário	MQTM	UserData

## Referência cruzada do cabeçalho ImqWork

Referência cruzada de atributos, estruturas de dados e campos para a classe C++ do Registro ImqAuthentication.

*Tabela 866. Mapeamento de atributos, estruturas de dados e campos*

Atributo	Estrutura de dados	Campo
message token (token de mensagem)	MQWIH	MessageToken
Nome do serviço	MQWIH	ServiceName
etapa de serviço	MQWIH	ServiceStep

## Classe C++ do Registro ImqAuthentication

Essa classe encapsula um registro de informações sobre autenticação (MQAIR) para uso durante a execução do método ImqQueueManager: :connect, para conexões do cliente TLS customizado.

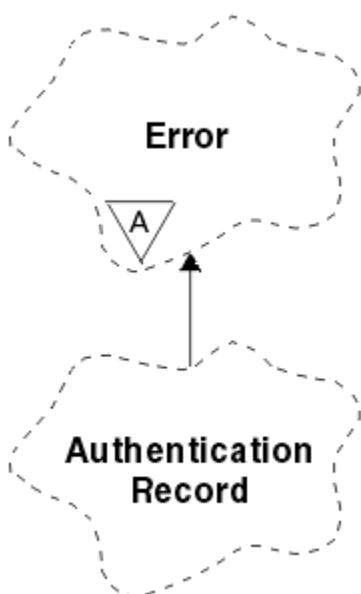


Figura 14. Classe de Registro ImqAuthentication

Consulte a descrição do método `ImqQueueManager::connect` para obter mais detalhes. Essa classe não está disponível para a plataforma z/OS

- [“Atributos do Objeto” na página 1866](#)
- [“Construtores” na página 1866](#)
- [“Métodos de objeto \(público\)” na página 1866](#)
- [“Métodos de objeto \(protegidos\)” na página 1867](#)

## Atributos do Objeto

### nome de conexão

O nome da conexão com o servidor CRL LDAP. Este é o endereço IP ou nome DNS, seguido opcionalmente pelo número da porta, entre parênteses.

### referência de conexão

Uma referência a um objeto do Gerenciador de `ImqQueue` que fornece a conexão necessária para um gerenciador de filas (local) O valor inicial é zero. Não confunda isso com o nome do gerenciador de filas que identifica um gerenciador de filas (possivelmente remoto) para uma fila nomeada..

### próximo registro de autenticação

Próximo objeto dessa classe, em nenhuma ordem específica, que tenha a mesma **referência de conexão** que esse objeto O valor inicial é zero.

### senha

Uma senha fornecida para a autenticação de conexão com o servidor LDAP CRL

### registro de autenticação anterior

Objeto anterior dessa classe, sem ordem específica, tendo a mesma **referência de conexão** que esse objeto. O valor inicial é zero.

### Tipo

O tipo de informações sobre autenticação contidas no registro

### nome do usuário

Um identificador de usuário fornecido para autorização para o servidor CRL LDAP.

## Construtores

### `ImqAuthenticationRecord ()`;

O construtor padrão..

## Métodos de objeto (público)

### `void operator = (const ImqAuthenticationRecord & air)`;

Copia os dados de instância do *air*, substituindo os dados de instância existentes

### `const ImqString e connectionName () const`;

Retorna o **nome da conexão**

### `void setConnectionNome (const ImqString & name)`;

Configura o **nome da conexão**

### `void setConnectionName (const char * name = 0)`;

Configura o **nome da conexão**

### `ImqQueueGerenciador * connectionReference () const`;

Retorna a **referência de conexão**

### `void setConnectionReferência ( ImqQueueManager & manager )`;

Configura a **referência de conexão**

### `void setConnectionReferência ( ImqQueueManager * manager = 0)`;

Configura a **referência de conexão**

**void copyOut (MQAIR \* pAir);**

Copia os dados de instância para *pAir*, substituindo os dados de instância existentes. Isso pode envolver alocar armazenamento dependente.

**void clear (MQAIR \* pAir);**

Limpa a estrutura e libera o armazenamento dependente referido por *pAir*.

**ImqAuthenticationRegistro \* nextAuthenticationRegistro () const;**

Retorna o **próximo registro de autenticação**

**const ImqString & senha () const;**

Retorna a **senha**

**void setPassword (const ImqString e password);**

Configura a **senha**

**void setPassword (const char \* password = 0);**

Configura a **senha**

**ImqAuthenticationRegistro \* previousAuthenticationRegistro () const;**

Retorna o **registro de autenticação anterior**

**Tipo MQLONG () const;**

Retorna o **type..**

**void setType (const MQLONG type);**

Configura o **tipo**

**const ImqString & userName () const;**

Retorna o **nome do usuário**

**void setUsername (const ImqString & name); e**

Configura o **nome do usuário**

**void setUsername (const char \* name = 0);**

Configura o **nome do usuário**

## Métodos de objeto (protegidos)

**void setNextAuthenticationRecord ( ImqAuthenticationRecord \* pAir = 0);**

Configura o **próximo registro de autenticação**

**Atenção:** Use esta função somente se você tiver certeza de que ela não quebrará a lista de registro de autenticação

**void setPreviousAuthenticationRecord ( ImqAuthenticationRecord \* pAir = 0);**

Configura o **registro de autenticação anterior**

**Atenção:** Use esta função somente se você tiver certeza de que ela não quebrará a lista de registro de autenticação

## Classe C++ ImqBinary

Essa classe encapsula uma matriz de bytes binários que pode ser usada para os valores de *ImqMessage token de contabilidade, correlation id e message id*. Ele permite fácil atribuição, cópia e comparação.

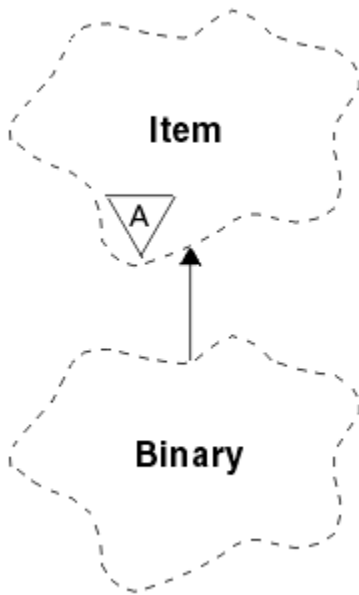


Figura 15. Classe *ImqBinary*

- [“Atributos do Objeto”](#) na página 1868
- [“Construtores”](#) na página 1868
- [“Métodos \*ImqItem\* sobrecarregados”](#) na página 1868
- [“Métodos de objeto \(público\)”](#) na página 1869
- [“Métodos de objeto \(protegidos\)”](#) na página 1869
- [“Códigos de Razão”](#) na página 1869

## Atributos do Objeto

### **data**

Uma matriz de bytes de dados binários O valor inicial é nulo.

### **Comprimento dos Dados**

O número de bytes.. O valor inicial é zero.

### **ponteiro de dados**

O endereço do primeiro byte dos **dados** O valor inicial é zero.

## Construtores

### ***ImqBinary*( );**

O construtor padrão..

### ***ImqBinary*( const *ImqBinary* & *binary* );**

O construtor de cópia..

### ***ImqBinary*( const void \* *data*, const size\_t *length* );**

Copia *bytes de comprimento* de *dados*

## Métodos *ImqItem* sobrecarregados

### **virtual *ImqBoolean* copyOut ( *ImqMessage* & *msg* );**

Copia os **dados** para o buffer de mensagem, substituindo qualquer conteúdo existente Configura o *msg format* para MQFMT\_NONE

Consulte a descrição do método de classe *ImqItem* , para obter detalhes adicionais



**virtual ImqBoolean pasteIn ( ImqMessage & msg );**

Configura os **dados** transferindo os dados restantes do buffer de mensagem, substituindo os **dados** existentes.

Para ser bem-sucedido, o ImqMessage **format** deve ser MQFMT\_NONE..

Consulte a descrição do método de classe ImqItem , para obter detalhes adicionais

**Métodos de objeto (público)****void operator = ( const ImqBinary e binary );**

Copia bytes de *binary*..

**ImqBoolean operator == ( const ImqBinary & binary );**

Compara esse objeto com *binário* Caso contrário, retornará FALSE se não for igual e TRUE. Os objetos serão iguais se tiverem o mesmo **comprimento de dados** e os bytes correspondentes.

**ImqBoolean copyOut ( void \* buffer, const size\_t length, const char pad = 0);**

Copia até *length* bytes do **ponteiro de dados** para *buffer*.. Se o **comprimento de dados** for insuficiente, o espaço restante no *buffer* será preenchido com *pad* bytes. *buffer* pode ser zero se *length* também for zero. O *length* não deve ser negativo Ele retornará TRUE se for bem-sucedido

**size\_t dataLength () const ;**

Retorna o **comprimento de dados**..

**ImqBoolean setDataLength ( const size\_t length );**

Configura o **comprimento de dados** Se o **comprimento de dados** for mudado como resultado desse método, os dados no objeto não serão inicializados Ele retornará TRUE se for bem-sucedido

**void \* dataPointer () const ;**

Retorna o **ponteiro de dados**

**ImqBoolean isNull () const ;**

Retorna TRUE se o **comprimento dos dados** for zero ou se todos os **dados** bytes forem zero. Caso contrário, retorna FALSE.

**ImqBoolean set ( const void \* buffer, const size\_t length );**

Copia *length* bytes de *buffer*. Ele retornará TRUE se for bem-sucedido

**Métodos de objeto (protegidos)****void clear ();**

Reduz o **comprimento de dados** para zero.

**Códigos de Razão**

- MQRC\_NO\_BUFFER
- MQRC\_STORAGE\_NOT\_AVAILABLE
- MQRC\_INCONSISTENT\_FORMAT

**Classe C++ ImqCache**

Use esta classe para reter ou serializar dados na memória

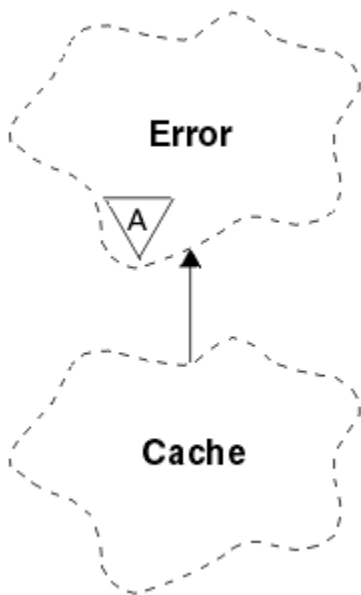


Figura 16. Classe *ImqCache*

Use esta classe para reter ou serializar dados na memória. É possível nomear um buffer de memória de tamanho fixo ou o sistema pode fornecer uma quantidade flexível de memória automaticamente. Essa classe está relacionada às chamadas MQI listadas em [“Referência cruzada de ImqCache”](#) na página 1850.

- [“Atributos do Objeto”](#) na página 1870
- [“Construtores”](#) na página 1871
- [“Métodos de objeto \(público\)”](#) na página 1871
- [“Códigos de Razão”](#) na página 1872

## Atributos do Objeto

### buffer automático

Indica se a memória buffer é gerenciada automaticamente pelo sistema (TRUE) ou é fornecida pelo usuário (FALSE). Ele é configurado inicialmente como TRUE

Esse atributo não é configurado diretamente. Ele é configurado indiretamente usando o método **useEmptyBuffer** ou **useFullBuffer**.

Se o armazenamento do usuário for fornecido este atributo é FALSE, a memória de buffer não pode crescer e erros de estouro de buffer podem ocorrer. O endereço e comprimento do buffer permanecem constantes.

Se o armazenamento do usuário não for fornecido, esse atributo será TRUE, e a memória de buffer poderá crescer incrementalmente para acomodar uma quantidade arbitrária dos dados da mensagem. No entanto, quando o buffer crescer, o endereço do buffer poderá ser alterado, portanto, tenha cuidado ao usar o **ponteiro do buffer** e o **ponteiro de dados**.

### Comprimento do buffer

O número de bytes de memória no buffer. O valor inicial é zero.

### ponteiro do buffer

O endereço da memória de buffer. O valor inicial é nulo.

### Comprimento dos Dados

O número de bytes seguintes ao **ponteiro de dados**. Deve ser igual ou menor que o **comprimento da mensagem**. O valor inicial é zero.

### Deslocamento de dados

O número de bytes anteriores ao **ponteiro de dados**. Deve ser igual ou menor que o **comprimento da mensagem**. O valor inicial é zero.

## ponteiro de dados

O endereço da parte do buffer que deve ser gravado ou lido a partir do próximo. O valor inicial é nulo.

## comprimento da mensagem

O número de bytes de dados significativos no buffer.. O valor inicial é zero.

## Construtores

### **ImqCache();**

O construtor padrão..

### **ImqCache( const ImqCache & cache );**

O construtor de cópia..

## Métodos de objeto (público)

### **void operator = ( const ImqCache & cache );**

Copia até **message length** bytes de dados do objeto *cache* para o objeto. Se **buffer automático** for FALSE, o **buffer length** já deverá ser suficiente para acomodar os dados copiados.

### **ImqBoolean automaticBuffer () const ;**

Retorna o valor **buffer automático** .

### **size\_t bufferLength () const ;**

Retorna o **buffer length**.

### **char \* bufferPointer () const ;**

Retorna o **ponteiro do buffer**

### **void clearMessage ();**

Configura o **comprimento da mensagem** e **deslocamento de dados** para zero.

### **size\_t dataLength () const ;**

Retorna o **comprimento de dados**..

### **size\_t dataOffset () const ;**

Retorna o **deslocamento de dados**

### **ImqBoolean setDataOffset ( const size\_t offset );**

Configura o **deslocamento de dados** O **comprimento da mensagem** é aumentado se necessário para assegurar que ele não seja menor que o **deslocamento de dados**. Esse método retornará TRUE se for bem-sucedido

### **char \* dataPointer () const ;**

Retorna uma cópia do **ponteiro de dados**

### **size\_t messageLength () const ;**

Retorna o **comprimento da mensagem**..

### **ImqBoolean setMessageLength ( const size\_t length );**

Configura o **comprimento da mensagem** Aumenta o **comprimento do buffer** se necessário para assegurar que o **comprimento da mensagem** não seja maior que o **comprimento do buffer**. Reduz o **deslocamento de dados** se necessário para assegurar que ele não seja maior que o **comprimento da mensagem**. Ele retornará TRUE se for bem-sucedido

### **ImqBoolean moreBytes ( const size\_t bytes-required );**

Assegura que *bytes-required* mais bytes estão disponíveis (para gravação) entre o **ponteiro de dados** e o final do buffer. Ele retornará TRUE se for bem-sucedido

Se **buffer automático** for TRUE, mais memória será adquirida conforme necessário; caso contrário, o **comprimento do buffer** já deverá ser adequado.

### **ImqBoolean read ( const size\_t length, char \* & external-buffer );**

Copia *length* bytes, do buffer iniciando na posição **ponteiro de dados** , no *external-buffer*. Após os dados serem copiados, o **deslocamento de dados** é aumentado em *length*. Esse método retornará TRUE se for bem-sucedido

### **ImqBoolean resizeBuffer ( const size\_t length );**

Varia o **comprimento do buffer**, desde que o **buffer automático** seja TRUE Isso é obtido realocando a memória do buffer. Até **comprimento da mensagem** bytes de dados do buffer existente são copiados para o novo. O número máximo copiado é de *comprimento* bytes O **ponteiro do buffer** é alterado O **comprimento da mensagem** e **deslocamento de dados** são preservados o mais próximo possível dos limites do novo buffer. Ele retorna TRUE se for bem-sucedido e FALSE se **buffer automático** for FALSE.

**Nota:** Esse método pode falhar com MQRC\_STORAGE\_NOT\_AVAILABLE se houver qualquer problema com os recursos do sistema

### **ImqBoolean useEmptyBuffer ( const char \* external-buffer, const size\_t length );**

Identifica um buffer do usuário vazio, configurando o **ponteiro do buffer** para apontar para *buffer externo*, o **comprimento do buffer** para *comprimento* e o **comprimento da mensagem** para zero.. Executa um **clearMessage**.. Se o buffer estiver totalmente preparado com dados, use o método **useFullBuffer** . Se o buffer tiver sido parcialmente preparado com dados, use o método **setMessageLength** para indicar a quantia correta Esse método retornará TRUE se for bem-sucedido

Esse método pode ser usado para identificar uma quantia fixa de memória, conforme descrito anteriormente ( *buffer externo* não é nulo e *comprimento* é diferente de zero); nesse caso, **buffer automático** é configurado como FALSE ou pode ser usado para reverter para a memória flexível gerenciada pelo sistema ( *buffer externo* é nulo e *comprimento* é zero); nesse caso, **buffer automático** é configurado como TRUE.

### **ImqBoolean useFullBuffer ( const char \* externalBuffer, const size\_t length );**

Como para **useEmptyBuffer**, exceto que o **comprimento da mensagem** é configurado como *comprimento* Ele retornará TRUE se for bem-sucedido

### **ImqBoolean write ( const size\_t length, const char \* external-buffer );**

Copia *length* bytes, do *external-buffer*, para o buffer iniciando na posição **ponteiro de dados** . Após os dados terem sido copiados, o **deslocamento de dados** é aumentado em *length* e o **message length** é aumentado, se necessário, para assegurar que não seja menor que o novo valor de **deslocamento de dados** . Esse método retornará TRUE se for bem-sucedido

Se o **buffer automático** for TRUE, uma quantidade adequada de memória será garantida; caso contrário, o **deslocamento de dados** final não deverá exceder o **comprimento do buffer**

## **Códigos de Razão**

- MQRC\_BUFFER\_NOT\_AUTOMATIC
- MQRC\_DATA\_TRUNCATED
- MQRC\_INSUFENT\_BUFFER
- MQRC\_INSUFENT\_DATA
- MQRC\_NULL\_POINTER
- MQRC\_STORAGE\_NOT\_AVAILABLE
- MQRC\_ZERO\_LENGTH

## **Classe C++ ImqChannel**

Essa classe encapsula uma definição de canal (MQCD) para uso durante a execução do método Manager: :connect, para conexões customizadas do cliente...

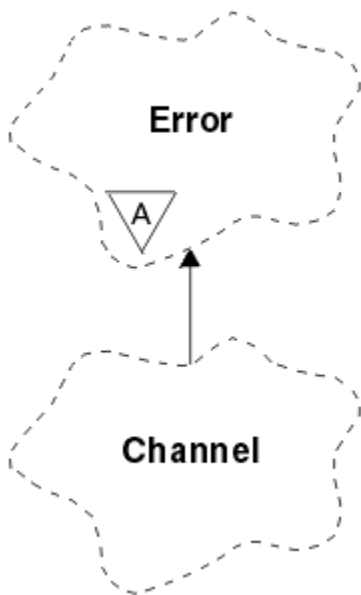


Figura 17. Classe *ImqChannel*

Consulte a descrição do método `Manager::connect` e [Programa de amostra HELLO WORLD \(imqwrlld.cpp\)](#) para obter mais detalhes.

Nem todos os métodos listados são aplicáveis a todas as plataformas. Consulte as descrições dos comandos [DEFINE CHANNEL](#) e [ALTER CHANNEL](#) para obter mais informações.

A classe `ImqChannel` não é suportada em z/OS..

- [“Atributos do Objeto”](#) na página 1873
- [“Construtores”](#) na página 1874
- [“Métodos de objeto \(público\)”](#) na página 1874
- [“Códigos de Razão”](#) na página 1878

## Atributos do Objeto

### batimento cardíaco em lote

O número de milissegundos entre verificações de que um canal remoto está ativo. O valor inicial é 0.

### nome do canal

O nome do canal. O valor inicial é nulo.

### nome de conexão

O nome da conexão. Por exemplo, o endereço IP de um computador host. O valor inicial é nulo.

### Compactação do cabeçalho

A lista de técnicas de compactação de dados de cabeçalho suportadas pelo canal. Os valores iniciais são todos configurados como `MQCOMPRESS_NOT_AVAILABLE`

### intervalo de batimento cardíaco

O número de segundos entre verificações que uma conexão ainda está funcionando. O valor inicial é 300.

### Intervalo keep-alive

O número de segundos passados para a pilha de comunicações especificando a sincronização de keep alive para o canal. O valor inicial é `MQKAI_AUTO`.

### endereço local

O endereço de comunicação local do canal.

**comprimento máximo da mensagem**

O comprimento máximo da mensagem suportada pelo canal em uma única comunicação. O valor inicial é 4 194 304.

**Compressão da mensagem**

A lista de técnicas de compactação de dados da mensagem suportada pelo canal. Os valores iniciais são todos configurados como MQCOMPRESS\_NOT\_AVAILABLE

**nome de modo**

O nome do modo. O valor inicial é nulo.

**senha**

Uma senha fornecida para a autenticação de conexão O valor inicial é nulo.

**contagem de saída de recebimento**

O número de saídas recebidas. O valor inicial é zero. Esse atributo é somente leitura.

**nomes de saída de recebimento**

Os nomes das saídas de recebimento

**receber dados do usuário**

Dados associados a saídas de recebimento

**Nome da saída de segurança**

O nome de uma saída de segurança a ser chamado no lado do servidor da conexão.. O valor inicial é nulo.

**dados do usuário de segurança**

Dados a serem transmitidos à saída de segurança. O valor inicial é nulo.

**contagem de saída de envio**

O número de saídas de envio O valor inicial é zero. Esse atributo é somente leitura.

**enviar nomes de saída**

Os nomes das saídas de envio

**enviar dados do usuário**

Dados associados a saídas de envio..

**SSL CipherSpec**

CipherSpec para uso com o TLS

**tipo de autenticação de cliente SSL**

O tipo de autenticação de cliente para uso com TLS

**Nome do peer SSL**

Nome do peer para uso com o TLS

**nome do programa de transação**

O nome do programa de transação.. O valor inicial é nulo.

**tipo de transporte**

O tipo de transporte da conexão. O valor inicial é MQXPT\_LU62.

**ID do usuário**

Um identificador de usuário fornecido para autorização. O valor inicial é nulo.

**Construtores****ImqChannel( ) ;**

O construtor padrão..

**ImqChannel( const ImqChannel & canal );**

O construtor de cópia..

**Métodos de objeto (público)****operador void = (const ImqChannel & channel );**

Copia dados de instância do *canal*, substituindo quaisquer dados de instância existentes.

**MQLONG batchHeartBeat () const;**

Retorna a **pulsação em lote**.

**ImqBoolean setBatchHeartBeat(const MQLONG *pulsação* = 0L );**

Configura a **pulsação em lote**. Esse método retornará TRUE se for bem-sucedido

**ImqString channelName() const;**

Retorna o **nome do canal**.

**ImqBoolean setChannelNome (const char \* *name* = 0);**

Configura o **nome do canal**. Esse método retornará TRUE se for bem-sucedido

**ImqString connectionName() const;**

Retorna o **nome da conexão**

**ImqBoolean setConnectionNome (const char \* *name* = 0);**

Configura o **nome da conexão** Esse método retornará TRUE se for bem-sucedido

**size\_t headerCompressionCount () const;**

Retorna a contagem de técnicas de compressão de dados de cabeçalho suportadas

**ImqBoolean headerCompression(const size\_t *count*, MQLONG *compress* []) const;**

Retorna cópias das técnicas de compactação de dados de cabeçalho suportadas em **compress** Esse método retornará TRUE se for bem-sucedido

**ImqBoolean setHeaderCompactação (const size\_t *count*, const MQLONG *compress* []);**

Configura as técnicas de compactação de dados do cabeçalho suportadas para **compactar**

Configura a contagem de técnicas de compactação de dados de cabeçalho suportadas para **count**

Esse método retornará TRUE se for bem-sucedido

**MQLONG heartBeatInterval () const;**

Retorna o **intervalo de pulsação**.

**ImqBoolean setHeartBeatInterval(const MQLONG *interval* = 300L );**

Configura o **intervalo de pulsação**. Esse método retornará TRUE se for bem-sucedido

**Intervalo de MQLONG keepAlive() const;**

Retorna o **intervalo de keep alive**.

**ImqBoolean setKeepAliveInterval(const MQLONG *interval* = MQKAI\_AUTO);**

Configura o **intervalo de keep alive**. Esse método retornará TRUE se for bem-sucedido

**ImqString localAddress() const;**

Retorna o **endereço local**..

**ImqBoolean setLocalAddress (const char \* *address* = 0);**

Configura o **endereço local** Esse método retornará TRUE se for bem-sucedido

**MQLONG maximumMessageLength () const;**

Retorna o **comprimento máximo da mensagem**..

**ImqBoolean setMaximumMessageLength(const MQLONG *length* = 4194304L );**

Configura o **comprimento máximo da mensagem**.. Esse método retornará TRUE se for bem-sucedido

**size\_t messageCompressionCount () const;**

Retorna a contagem de técnicas de compressão de dados de mensagens suportadas

**ImqBoolean messageCompression(const size\_t *count*, MQLONG *compress* []) const;**

Retorna cópias das técnicas de compactação de dados da mensagem suportadas em **compress** Esse método retornará TRUE se for bem-sucedido

**ImqBoolean setMessageCompactação (const size\_t *count*, const MQLONG *compress* []);**

Configura as técnicas de compactação de dados da mensagem suportadas para compactação

Configura a contagem de técnicas de compactação de dados da mensagem suportadas para contagem

Esse método retornará TRUE se for bem-sucedido

**ImqString modeName() const;**

Retorna o **nome do modo**

**ImqBoolean setModeNome (const char \* name = 0);**

Configura o **nome do modo** Esse método retornará TRUE se for bem-sucedido

**ImqString senha () const;**

Retorna a **senha**

**ImqBoolean setPassword(const char \* password = 0);**

Configura a **senha** Esse método retornará TRUE se for bem-sucedido

**size\_t receiveExitCount () const;**

Retorna a **contagem de saída de recebimento**

**ImqString receiveExitName ();**

Retorna o primeiro dos **nomes de saída de recebimento**, se houver. Se a **contagem de saída de recebimento** for zero, ela retornará uma sequência vazia.

**ImqBoolean receiveExitNomes (const size\_t count, ImqString \* names []);**

Retorna cópias dos **nomes de saída de recebimento** em *names* Configura quaisquer *names* em excesso de **contagem de saída de recebimento** para sequências nulas. Esse método retornará TRUE se for bem-sucedido

**ImqBoolean setReceiveExitName(const char \* name = 0);**

Configura os **nomes de saída de recebimento** para o único *nome name* pode ser em branco ou nulo. Configura a **contagem de saídas de recebimento** para 1 ou zero. Limpa os **dados do usuário de recebimento** Esse método retornará TRUE se for bem-sucedido

**ImqBoolean setReceiveExitNames(const size\_t count, const char \* names []);**

Configura os **nomes de saída de recebimento** para *names* Os valores de *names* individuais não devem estar em branco ou ser nulos. Configura a **contagem de saída de recebimento** para *contagem* Limpa os **dados do usuário de recebimento** Esse método retornará TRUE se for bem-sucedido

**ImqBoolean setReceiveExitNames(const size\_t count, const ImqString \* names []);**

Configura os **nomes de saída de recebimento** para *names* Os valores de *names* individuais não devem estar em branco ou ser nulos. Configura a **contagem de saída de recebimento** para *contagem* Limpa os **dados do usuário de recebimento** Esse método retornará TRUE se for bem-sucedido

**ImqString receiveUserDados ();**

Retorna o primeiro dos itens **receber dados do usuário**, se houver. Se a **contagem de saída de recebimento** for zero, retornará uma sequência vazia.

**ImqBoolean receiveUserDados (const size\_t count, ImqString \* data []);**

Retorna cópias dos itens **receber dados do usuário** em *dados*. Configura quaisquer *dados* em excesso de **receber contagem de saída** para sequências nulas. Esse método retornará TRUE se for bem-sucedido

**ImqBoolean setReceiveUserData(const char \* data = 0);**

Configura **receber dados do usuário** para o item único *dados*. Se *data* não for nulo, **contagem de saída de recebimento** deverá ser pelo menos 1. Esse método retornará TRUE se for bem-sucedido

**ImqBoolean setReceiveUserData(const size\_t count, const char \* data []);**

Configura **receber dados do usuário** para *dados*. *count* não deve ser maior que **receive exit count**. Esse método retornará TRUE se for bem-sucedido

**ImqBoolean setReceiveUserData(const size\_t count, const ImqString \* data []);**

Configura **receber dados do usuário** para *dados*. *count* não deve ser maior que **receive exit count**. Esse método retornará TRUE se for bem-sucedido

**ImqString securityExitNome () const;**

Retorna o **nome da saída de segurança**

**ImqBoolean setSecurityExitName(const char \* name = 0);**

Configura o **nome da saída de segurança** Esse método retornará TRUE se for bem-sucedido

**ImqString securityUserDados () const;**

Retorna os **dados do usuário de segurança**



**ImqBoolean setSecurityUserData(const char \* data = 0);**

Configura os **dados do usuário de segurança** Esse método retornará TRUE se for bem-sucedido

**size\_t sendExitContagem () const;**

Retorna a **contagem de saída de envio**

**ImqString sendExitName ();**

Retorna o primeiro dos **nomes de saída de envio**, se houver. Retorna uma sequência vazia se a **contagem de saída de envio** for zero.

**ImqBoolean sendExitNomes (const size\_t count, ImqString \* names []);**

Retorna cópias do **send exit names** em *names*. Configura quaisquer *nomes* além de **enviar contagem de saída** para sequências nulas. Esse método retornará TRUE se for bem-sucedido

**ImqBoolean setSendExitName(const char \* name = 0);**

Configura os **nomes de saída de envio** para o único *nome* *name* pode ser em branco ou nulo. Configura a **contagem de saída de envio** para 1 ou zero. Limpa os **dados do usuário de envio** Este método retorna TRUE se for bem-sucedido

**ImqBoolean setSendExitNames(const size\_t count, const char \* names []);**

Configura os **nomes de saída de envio** para *nomes* Os valores de *nomes* individuais não devem estar em branco ou ser nulos. Configura a **contagem de saída de envio** para *contagem* Limpa os **dados do usuário de envio** Esse método retornará TRUE se for bem-sucedido

**ImqBoolean setSendExitNames(const size\_t count, const ImqString \* names []);**

Configura os **nomes de saída de envio** para *nomes* Os valores de *nomes* individuais não devem estar em branco ou ser nulos. Configura a **contagem de saída de envio** para *contagem* Limpa os **dados do usuário de envio** Esse método retornará TRUE se for bem-sucedido

**ImqString sendUserDados ();**

Retorna o primeiro dos itens **enviar dados do usuário**, se houver., Retorna uma sequência vazia se a **contagem de saída de envio** for zero.

**ImqBoolean sendUserData (const size\_t count, ImqString \* data []);**

Retorna cópias dos itens **enviar dados do usuário** em *dados*. Configura quaisquer *dados* em excesso de **enviar contagem de saída** para sequências nulas. Esse método retornará TRUE se for bem-sucedido

**ImqBoolean setSendUserData(const char \* data = 0);**

Configura **enviar dados do usuário** para os *dados* de item único. Se *data* não for nulo, **contagem de saída de envio** deverá ser pelo menos 1. Esse método retornará TRUE se for bem-sucedido

**ImqBoolean setSendUserData(const size\_t count, const char \* data []);**

Configura **enviar dados do usuário** para *dados*. *count* não deve ser maior que **send exit count**. Esse método retornará TRUE se for bem-sucedido

**ImqBoolean setSendUserData(const size\_t count, const ImqString \* data []);**

Configura **enviar dados do usuário** para *dados*. *count* não deve ser maior que **send exit count**. Esse método retornará TRUE se for bem-sucedido

**ImqString sslCipherEspecificação () const;**

Retorna a especificação de cifra TLS

**ImqBoolean setSslCipherSpecification(const char \* name = 0);**

Configura a especificação de cifra TLS Esse método retornará TRUE se for bem-sucedido

**MQLONG sslClientAutenticação () const;**

Retorna o tipo de autenticação de cliente TLS..

**ImqBoolean setSslClientAuthentication(const MQLONG auth = MQSCA\_REQUIRED);**

Configura o tipo de autenticação de cliente TLS Esse método retornará TRUE se for bem-sucedido

**ImqString sslPeerNome () const;**

Retorna o nome do peer do TLS

**ImqBoolean setSslPeerName(const char \* name = 0);**

Configura o nome do peer do TLS Esse método retornará TRUE se for bem-sucedido

**ImqString transactionProgramNome () const;**

Retorna o **nome do programa de transação**

**ImqBoolean setTransactionProgramName(const char \* name = 0);**

Configura o **nome do programa de transação** Esse método retornará TRUE se for bem-sucedido

**MQLONG transportType() const;**

Retorna o **tipo de transporte**

**ImqBoolean setTransportTipo (const MQLONG tipo = MQXPT\_LU62 );**

Configura o **tipo de transporte** Esse método retornará TRUE se for bem-sucedido

**ImqString userId() const;**

Retorna o **ID do usuário**

**ImqBoolean setUserId (const char \* id = 0);**

Configura o **ID do usuário** Esse método retornará TRUE se for bem-sucedido

### Códigos de Razão

- MQRC\_DATA\_LENGTH\_ERROR
- MQRC\_ITEM\_COUNT\_ERROR
- MQRC\_NULL\_POINTER
- MQRC\_SOURCE\_BUFFER\_ERROR

## Classe C++ ImqCICSBridgeHeader

Essa classe contém recursos específicos da estrutura de dados MQCIH.

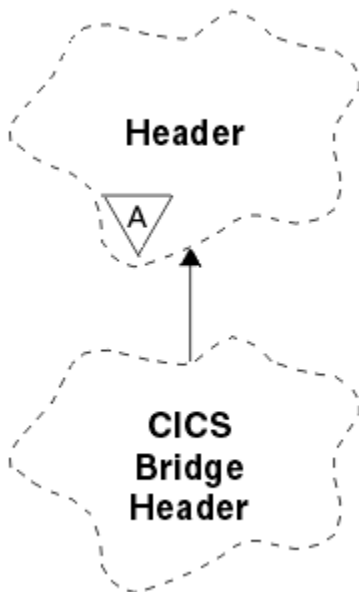


Figura 18. Classe ImqCICSBridgeHeader

Objetos dessa classe são usados por aplicativos que enviam mensagens para CICS bridge por meio de IBM MQ for z/OS.

- [“Atributos do Objeto” na página 1879](#)
- [“Construtores” na página 1881](#)
- [“Métodos ImqItem sobrecarregados” na página 1881](#)
- [“Métodos de objeto \(público\)” na página 1881](#)
- [“Dados do objeto \(protegidos\).” na página 1884](#)
- [“Códigos de Razão” na página 1884](#)
- [“Códigos de retorno” na página 1884](#)

## Atributos do Objeto

### Descritor ADS

Enviar / receber descritor ADS. Isso é configurado usando MQCADSD\_NONE O valor inicial é MQCADSD\_NONE. Os seguintes valores adicionais são possíveis:

- MQCADSD\_NONE
- MQCADSD\_SEND
- MQCADSD\_RECV
- MQCADSD\_MSGFORMAT

### identificador de atenção

Chave do AID. O campo deve ter o comprimento MQ\_ATTENTION\_ID\_LENGTH

### autenticador

Senha ou passticket do RACF O valor inicial contém espaços em branco, de comprimento MQ\_AUTHENTICATOR\_LENGTH.

### código de encerramento anormal de ponte

Código de encerramento anormal de ponte, de comprimento MQ\_ABEND\_CODE\_LENGTH. O valor inicial é quatro caracteres em branco. O valor retornado neste campo depende do código de retorno. Consulte [Tabela 867 na página 1884](#) para obter mais detalhes.

### código de cancelamento da ponte,

Código de transação de encerramento anormal da ponte. O campo é reservado, deve conter espaços em branco e ter comprimento MQ\_CANCEL\_CODE\_LENGTH.

### código de conclusão da ponte

Código de conclusão, que pode conter o código de conclusão IBM MQ ou o valor EIBRESP CICS . O campo tem o valor inicial de MQCC\_OK O valor retornado neste campo depende do código de retorno. Consulte [Tabela 867 na página 1884](#) para obter mais detalhes.

### compensação de erro de ponte

Deslocamento de erro de ponte O valor inicial é zero. Esse atributo é somente leitura.

### código de razão da ponte

Código de razão. Esse campo pode conter o motivo IBM MQ ou o valor CICS EIBRESP2 . O campo tem o valor inicial de MQRC\_NONE. O valor retornado neste campo depende do código de retorno. Consulte [Tabela 867 na página 1884](#) para obter mais detalhes.

### código de retorno da ponte

Código de retorno do CICS bridge. O valor inicial é MQCRC\_OK.

### tarefa conversacional

Se a tarefa pode ser de conversação O valor inicial é MQCCT\_NO.. Os seguintes valores adicionais são possíveis:

- MQCCT\_YES
- MQCCT\_NO

### posição do cursor

Posição do cursor. O valor inicial é zero.

### instalação manter tempo

Tempo de liberação da instalação do CICS bridge

### instalações como

Atributo emulado do terminal O campo deve ter comprimento MQ\_FACILITY\_LIKE\_LENGTH.

### token do recurso

Valor do token BVT. O campo deve ter o comprimento MQ\_FACILITY\_LENGTH O valor inicial é MQCFAC\_NONE.

### função

A função, que pode conter o nome da chamada IBM MQ ou a função EIBFN CICS O campo tem o valor inicial de MQCFUNC\_NONE, com comprimento MQ\_FUNCTION\_LENGTH. O valor retornado

neste campo depende do código de retorno. Consulte [Tabela 867 na página 1884](#) para obter mais detalhes.

Os seguintes valores adicionais são possíveis quando a **função** contém um nome de chamada IBM MQ :

- MQCFUNC\_MQCONN
- MQCFUNC\_MQGET
- MQCFUNC\_MQINQ
- MQCFUNC\_NONE
- MQCFUNC\_MQOPEN
- MQCFUNC\_PUT
- MQCFUNC\_MQPUT1

#### **obter intervalo de espera**

Intervalo de espera para uma chamada MQGET emitida pela tarefa CICS bridge . O valor inicial é MQCGWI\_DEFAULT. O campo se aplica apenas quando **uow control** possui o valor MQCUOWC\_FIRST. Os seguintes valores adicionais são possíveis:

- MQCGWI\_DEFAULT
- MQWI\_UNLIMITED

#### **tipo de link**

Tipo de link.. O valor inicial é MQCLT\_PROGRAM. Os seguintes valores adicionais são possíveis:

- MQCLT\_PROGRAM
- MQCLT\_TRANSACTION

#### **próximo identificador de transação**

ID da próxima transação a ser anexada O campo deve ter comprimento MQ\_TRANSACTION\_ID\_LENGTH.

#### **comprimento dos dados de saída**

Comprimento dos dados de COMMAREA O valor inicial é MQCODL\_AS\_INPUT.

#### **formato de resposta para**

O nome do formato da mensagem de resposta O valor inicial é MQFMT\_NONE com comprimento MQ\_FORMAT\_LENGTH.

#### **código de início**

Código inicial da transação. O campo deve ter comprimento MQ\_START\_CODE\_LENGTH. O valor inicial é MQCSC\_NONE. Os seguintes valores adicionais são possíveis:

- MQCSC\_START
- MQCSC\_STARTDATA
- MQCSC\_TERMININPUT
- MQCSC\_NONE

#### **status de encerramento da tarefa**

Status de término da tarefa O valor inicial é MQCTES\_NOSYNC. Os seguintes valores adicionais são possíveis:

- MQCTES\_COMMIT
- MQCTES\_BACKOUT
- MQCTES\_ENDTASK
- MQCTES\_NOSYNC

### identificador de transação

ID da transação a ser anexada O valor inicial deve conter espaços em branco e deve ter comprimento MQ\_TRANSACTION\_ID\_LENGTH. O campo se aplica apenas quando **uow control** tem o valor MQCUOWC\_FIRST ou MQCUOWC\_ONLY.

### Controle UOW

Controle de UOW O valor inicial é MQCUOWC\_ONLY. Os seguintes valores adicionais são possíveis:

- MQCUOWC\_FIRST
- MQCUOWC\_MIDDLE
- MQCUOWC\_LAST
- MQCUOWC\_ONLY
- MQCUOWC\_COMMIT
- MQCUOWC\_BACKOUT
- MQCUOWC\_CONTINUE

### versão

O número da versão do MQCIH O valor inicial é MQCIH\_VERSION\_2. O único outro valor suportado é MQCIH\_VERSION\_1.

## Construtores

### ImqCICSBridgeHeader();

O construtor padrão..

### ImqCICSBridgeHeader(const ImqCICSBridgeHeader & );

O construtor de cópia..

## Métodos ImqItem sobrecarregados

### virtual ImqBoolean copyOut( ImqMessage & msg );

Insere uma estrutura de dados MQCIH no buffer de mensagem no início, movendo os dados da mensagem existentes mais adiante e configura o formato da mensagem para MQFMT\_CICS

Consulte a descrição do método da classe-pai para obter mais detalhes

### virtual ImqBoolean pasteIn( ImqMessage & msg );

Lê uma estrutura de dados MQCIH do buffer de mensagem. Para ser bem sucedido, a codificação do objeto *msg* deve ser MQENC\_NATIVE Recuperar mensagens com MQGMO\_CONVERT para MQENC\_NATIVE. Para ser bem-sucedido, o formato ImqMessage deve ser MQFMT\_CICS..

Consulte a descrição do método da classe-pai para obter mais detalhes

## Métodos de objeto (público)

### operador void = (const ImqCICSBridgeHeader & header );

Copia os dados de instância do *header*, substituindo os dados de instância existentes

### MQLONG ADSDescriptor () const;

Retorna uma cópia do **descriptor ADS**.

### void setADSDescriptor(const MQLONG descriptor = MQCADSD\_NONE)

Configura o **descriptor ADS**.

### ImqString attentionIdentifier() const;

Retorna uma cópia do **identificador de atenção**, preenchido com espaços em branco finais para o comprimento MQ\_ATTENTION\_ID\_LENGTH.

### void setAttentionIdentificador (const char \* data = 0);

Configura o **identificador de atenção**, preenchido com espaços em branco finais para o comprimento MQ\_ATTENTION\_ID\_LENGTH. Se nenhum *dado* for fornecido, reconfigura **identificador de atenção** para o valor inicial.

**Autenticador ImqString () const;**

Retorna uma cópia do **autenticador**, preenchidos com espaços em branco à direita para o comprimento MQ\_AUTHENTICATOR\_LENGTH

**void setAuthenticator(const char \* data = 0)**

Configura o **autenticador**, preenchido com espaços em branco finais para comprimento MQ\_AUTHENTICATOR\_LENGTH. Se nenhum *dado* for fornecido, reconfigura **autenticador** para o valor inicial.

**ImqString bridgeAbendCódigo () const;**

Retorna uma cópia do **código de encerramento anormal da ponte**, preenchido com espaços em branco finais para o comprimento MQ\_ABEND\_CODE\_LENGTH.

**ImqString bridgeCancelCódigo () const;**

Retorna uma cópia do **código de cancelamento da ponte**, preenchido com espaços em branco à direita até o comprimento de MQ\_CANCEL\_CODE\_LENGTH

**void setBridgeCancelCode(const char \* data = 0);**

Configura o **código de cancelamento da ponte**, preenchido com espaços em branco à direita para o comprimento MQ\_CANCEL\_CODE\_LENGTH. Se nenhum *dado* for fornecido, reconfigura o **código de cancelamento de ponte** para o valor inicial.

**MQLONG bridgeCompletionCode () const;**

Retorna uma cópia do **código de conclusão da ponte**

**MQLONG bridgeErrorOffset () const;**

Retorna uma cópia do **deslocamento de erro de ponte**

**MQLONG bridgeReasonCódigo () const;**

Retorna uma cópia do **código de razão da ponte**

**Código MQLONG bridgeReturn() const;**

Retorna o **código de retorno da ponte**

**MQLONG conversationalTask() const;**

Retorna uma cópia da **tarefa de conversação**

**void setConversationalTask (const MQLONG task = MQCCT\_NO);**

Configura a **tarefa de conversação**

**MQLONG cursorPosition() const;**

Retorna uma cópia da **posição do cursor**

**void setCursorPosição (const MQLONG position = 0);**

Configura a **posição do cursor**

**MQLONG facilityKeepTime () const;**

Retorna uma cópia do **tempo de manutenção do recurso**

**anular setFacilityKeepTime(const MQLONG time = 0);**

Configura o **tempo de manutenção do recurso**

**ImqString facilityLike() const;**

Retorna uma cópia do recurso **como**, preenchidos com espaços em branco à direita para o comprimento MQ\_FACILITY\_LIKE\_LENGTH

**void setFacilityLike (const char \* name = 0);**

Configura o recurso **como**, preenchido com espaços em branco à direita para o comprimento MQ\_FACILITY\_LIKE\_LENGTH. Se nenhum *nome* for fornecido, reconfigura o recurso **como** o valor inicial.

**ImqBinary facilityToken() const;**

Retorna uma cópia do **token de recurso**

**ImqBoolean setFacilityToken (const ImqBinary & );**

Configura o **token do recurso**. O **comprimento de dados** do *token* deve ser zero ou MQ\_FACILITY\_LENGTH. Ele retornará TRUE se for bem-sucedido

**void setFacilityToken (const MQBYTE8 token = 0);**

Configura o **token do recurso** *token* pode ser zero, que é o mesmo que especificar MQCFAC\_NONE. Se *token* for diferente de zero, ele deverá endereçar bytes MQ\_FACILITY\_LENGTH de dados binários. Ao usar valores predefinidos, como MQCFAC\_NONE, pode ser necessário fazer uma conversão para assegurar uma correspondência de assinatura Por exemplo, (MQBYTE \*) MQCFAC\_NONE.

**Função ImqString () const;**

Retorna uma cópia da **função**, preenchidos com espaços em branco finais para o comprimento MQ\_FUNCTION\_LENGTH.

**Intervalo de MQLONG getWait() const;**

Retorna uma cópia do **get wait interval**.

**void setGetWaitInterval(const MQLONG interval = MQCGWI\_DEFA**

Configura o **get wait interval**.

**MQLONG linkType() const;**

Retorna uma cópia do **tipo de link**..

**void setLinkType (const MQLONG type = MQCLT\_PROGRAM);**

Configura o **tipo de vínculo**

**ImqString nextTransactionIdentificador () const;**

Retorna uma cópia dos dados do **próximo identificador de transação**, preenchidos com espaços em branco à direita para o comprimento MQ\_TRANSACTION\_ID\_LENGTH

**MQLONG outputDataLength () const;**

Retorna uma cópia do **comprimento de dados de saída**

**void setOutputDataLength(const MQLONG length = MQCODL\_AS\_INPUT);**

Configura o **comprimento de dados de saída**

**ImqString replyToFormat () const;**

Retorna uma cópia do nome do **reply-to format**, preenchido com espaços em branco à direita para o comprimento MQ\_FORMAT\_LENGTH

**void setReplyToFormat(const char \* name = 0);**

Configura o **formato de resposta**, preenchido com espaços em branco finais para o comprimento MQ\_FORMAT\_LENGTH. Se nenhum *name* for fornecido, reconfigurará **reply-to format** para o valor inicial.

**ImqString startCode() const;**

Retorna uma cópia do **código inicial**, preenchido com espaços em branco finais para o comprimento MQ\_START\_CODE\_LENGTH.

**void setStartCode (const char \* data = 0);**

Configura os dados do **código de início**, preenchidos com espaços em branco à direita para o comprimento MQ\_START\_CODE\_LENGTH Se nenhum *dado* for fornecido, reconfigura o **código inicial** para o valor inicial.

**MQLONG taskEndStatus () const;**

Retorna uma cópia do **status final da tarefa**.

**ImqString transactionIdentifier() const;**

Retorna uma cópia dos dados do **identificador de transação**, preenchidos com espaços em branco à direita para o comprimento MQ\_TRANSACTION\_ID\_LENGTH

**void setTransactionIdentifier (const char \* data = 0);**

Configura o **identificador de transação**, preenchido com espaços em branco à direita para o comprimento MQ\_TRANSACTION\_ID\_LENGTH. Se nenhum *dado* for fornecido, reconfigura **identificador de transação** para o valor inicial.

**MQLONG UOWControl () const;**

Retorna uma cópia do **controle de UOW**

**void setUOWControl(const MQLONG control = MQCUOWC\_ONLY);**

Configura o **Controle de UOW**

**MQLONG versão () const;**

Retorna o número da **versão**

**ImqBoolean setVersion(const MQLONG version = MQCIH\_VERSION\_2 );**

Configura o número da **versão** Ele retornará TRUE se for bem-sucedido

### Dados do objeto (protegidos).

#### **MQLONG olVersion**

O número máximo da versão do MQCIH que pode ser acomodado no armazenamento alocado para *opcih*

#### **PMQCIH opcih**

O endereço de uma estrutura de dados do MQCIH A quantidade de armazenamento alocado é indicada por *olVersion*

### Códigos de Razão

- MQRC\_BINARY\_DATA\_LENGTH\_ERROR
- MQRC\_ERRONE\_VERSION

### Códigos de retorno

*Tabela 867. Códigos de Retorno da Classe ImqCICSBridgeHeader*

<b>Código de Retorno</b>	<b>Função</b>	<b>CompCode</b>	<b>Razão</b>	<b>Código de aborto</b>
MQCRC_OK				
MQCRC_BRIDGE_ERROR			MQFB_CICS	
MQCRC_MQ_API_ERROR	Nome da chamada IBM MQ	IBM MQ CompCode	IBM MQ Motivo	
MQCRC_BRIDGE_TIMEOUT	Nome da chamada IBM MQ	IBM MQ CompCode	IBM MQ Motivo	
MQCRC_CICS_EXEC_ERROR	CICS EIBFN	CICS EIBRESP	CICS EIBRESP2	
MQCRC_SECURITY_ERROR	CICS EIBFN	CICS EIBRESP	CICS EIBRESP2	
MQCRC_PROGRAM_NOT_AVAILABLE	CICS EIBFN	CICS EIBRESP	CICS EIBRESP2	
MQCRC_TRANSID_NOT_AVAILABLE	CICS EIBFN	CICS EIBRESP	CICS EIBRESP2	
MQCRC_BRIDGE_ABEND				CICS ABCODE
MQCRC_APPLICATION_ABEND				CICS ABCODE

### ImqDeadLetterHeader Classe C++

Essa classe contém recursos da estrutura de dados MQDLH.



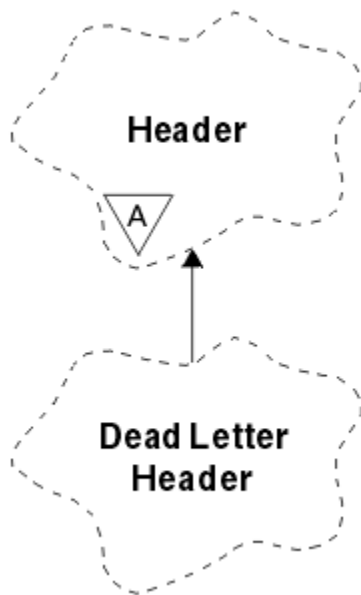


Figura 19. classe *ImqDeadLetterHeader*

Os objetos dessa classe geralmente são usados por um aplicativo que encontra uma mensagem que não pode ser processada. Uma nova mensagem que inclui um cabeçalho de mensagens não entregues e o conteúdo da mensagem é colocado na fila de mensagens não entregues e a mensagem é descartada.

- [“Atributos do Objeto” na página 1885](#)
- [“Construtores” na página 1886](#)
- [“Métodos \*ImqItem\* sobrecarregados” na página 1886](#)
- [“Métodos de objeto \(público\)” na página 1886](#)
- [“Dados do objeto \(protegidos\).” na página 1887](#)
- [“Códigos de Razão” na página 1887](#)

## Atributos do Objeto

### código de razão de devoluções

O motivo pelo qual a mensagem chegou à fila de mensagens não entregues. O valor inicial é `MQRC_NONE`.

### Nome do gerenciador de filas de destino

O nome do gerenciador de fila de destino original. O nome é uma sequência de comprimento `MQ_Q_MGR_NAME_LENGTH`. Seu valor inicial é nulo.

### Nome da Fila de Destino

O nome da fila de destino original. O nome é uma sequência de comprimento `MQ_Q_NAME_LENGTH`. Seu valor inicial é nulo.

### Colocar nome do aplicativo

O nome do aplicativo que colocou a mensagem na fila de devoluções. O nome é uma sequência de comprimento `MQ_PUT_APPL_NAME_LENGTH`. Seu valor inicial é nulo.

### Colocar tipo de aplicativo

O tipo de aplicativo que colocou a mensagem em uma fila de mensagens não entregues. O valor inicial é zero.

### Data de colocação

A data em que a mensagem foi colocada na fila de devoluções. A data é uma sequência de comprimento `MQ_PUT_DATE_LENGTH`. Seu valor inicial é uma cadeia nula.

## Horário de colocação

A hora em que a mensagem foi colocada na fila de devoluções. O horário é uma sequência de comprimento MQ\_PUT\_TIME\_LENGTH Seu valor inicial é uma cadeia nula.

## Construtores

### **ImqDeadLetterHeader( );**

O construtor padrão..

### **ImqDeadLetterHeader(const ImqDeadLetterHeader & header );**

O construtor de cópia..

## Métodos ImqItem sobrecarregados

### **virtual ImqBoolean copyOut ( ImqMessage & msg );**

Insere uma estrutura de dados MQDLH no buffer de mensagem no início, movendo os dados da mensagem existente mais adiante Configura o formato de *msg* para MQFMT\_DEAD\_LETTER\_HEADER

Consulte a descrição do método de classe ImqHeader na página [“Classe C++ ImqHeader” na página 1893](#) para obter detalhes adicionais

### **virtual ImqBoolean pasteIn ( ImqMessage & msg );**

Lê uma estrutura de dados MQDLH do buffer de mensagem.

Para ser bem-sucedido, o formato ImqMessage deve ser MQFMT\_DEAD\_LETTER\_HEADER..

Consulte a descrição do método de classe ImqHeader na página [“Classe C++ ImqHeader” na página 1893](#) para obter detalhes adicionais

## Métodos de objeto (público)

### **void operator = (const ImqDeadLetterHeader & header );**

Copia os dados de instância que são copiados do *header*, substituindo os dados de instância existentes

### **MQLONG deadLetterReasonCode () const;**

Retorna o código de razão de devoluções

### **void setDeadLetterReasonCode (const MQLONG reason );**

Configura o código de razão de devoluções

### **ImqString destinationQueueManagerName (). const;**

Retorna o nome do gerenciador de filas de destino, removido de quaisquer espaços em branco finais.

### **void setDestinationQueueManagerNome (const char \* name )**

Configura o nome do gerenciador de filas de destino Trunca dados maiores que MQ\_Q\_MGR\_NAME\_LENGTH (48 caracteres).

### **ImqString destinationQueueNome () const;**

Retorna uma cópia do nome da fila de destino, removida de quaisquer espaços em branco finais.

### **void setDestinationQueueName (const char \* name );**

Define o nome da fila de destino Trunca dados maiores que MQ\_Q\_NAME\_LENGTH (48 caracteres).

### **ImqString putApplicationNome () const;**

Retorna uma cópia do nome do aplicativo put, removido de quaisquer espaços em branco finais.

### **void setPutApplicationName (const char \* name = 0);**

Configura o nome do aplicativo de colocação Trunca dados mais longos que MQ\_PUT\_APPL\_NAME\_LENGTH (28 caracteres).

### **MQLONG putApplicationTipo () const;**

Retorna o tipo de aplicativo de colocação

### **void setPutApplicationType (const MQLONG type = MQAT\_NO\_CONTEXT);**

Configura o tipo de aplicativo de colocação

**ImqString putDate () const;**

Retorna uma cópia da data de colocação, removida de quaisquer espaços em branco à direita.

**void setPutDate (const char \* date = 0);**

Configura a data de colocação Trunca dados mais longos que MQ\_PUT\_DATE\_LENGTH (8 caracteres).

**ImqString putTime () const;**

Retorna uma cópia do tempo de colocação, despido de quaisquer espaços em branco à direita.

**void setPutTime (const char \* time = 0);**

Configura o tempo de colocação Trunca dados maiores que MQ\_PUT\_TIME\_LENGTH (8 caracteres).

**Dados do objeto (protegidos).****MQDLH omqdlh**

A estrutura de dados do MQDLH

**Códigos de Razão**

- MQRC\_INCONSISTENT\_FORMAT
- MQRC\_STRUC\_ID\_ERROR
- MQRC\_ENCODING\_ERROR

**ImqDistributionListar classe C++**

Esta classe encapsula uma lista de distribuição dinâmica que faz referência a uma ou mais filas para o propósito de enviar uma mensagem ou mensagens para vários destinos

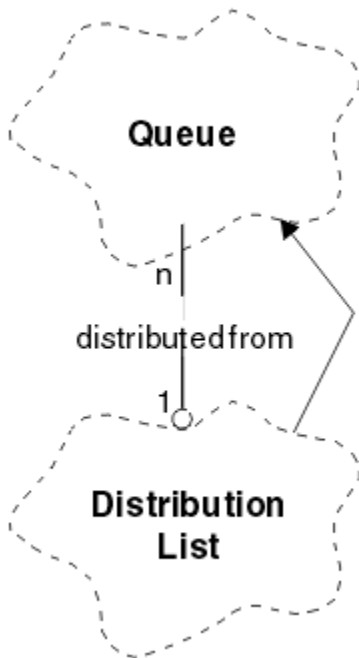


Figura 20. Classe de lista ImqDistribution.

- [“Atributos do Objeto” na página 1888](#)
- [“Construtores” na página 1888](#)
- [“Métodos de objeto \(público\)” na página 1888](#)
- [“Métodos de objeto \(protegidos\)” na página 1888](#)

## Atributos do Objeto

### primeira fila distribuída

O primeiro de um ou mais objetos de classe, em nenhuma ordem específica, no qual a **referência da lista de distribuições** aborda esse objeto

Inicialmente não existem tais objetos. Para abrir uma Lista de ImqDistribution com êxito, deve haver pelo menos um objeto desse tipo

**Nota:** Quando um objeto de Lista ImqDistribution é aberto, quaisquer objetos abertos que fazem referência a ele são automaticamente fechados

## Construtores

### ImqDistributionList ();

O construtor padrão..

### ImqDistributionList ( const ImqDistributionList & lista );

O construtor de cópia..

## Métodos de objeto (público)

### void operator = ( const ImqDistributionList e list );

Todos os objetos que referenciam **esse** objeto são desreferenciados antes da cópia Nenhum objeto fará referência a **este** objeto após a chamada deste método

### \* firstDistributedFila () const ;

Retorna a **primeira fila distribuída**..

## Métodos de objeto (protegidos)

### void setFirstDistributedQueue ( \* queue = 0);

Configura a **primeira fila distribuída**

## Classe C++ ImqError

Esta classe abstrata fornece informações sobre os erros associados a um objeto

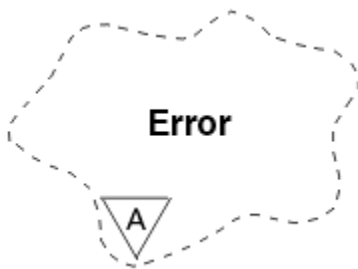


Figura 21. Classe ImqError

- [“Atributos do Objeto” na página 1888](#)
- [“Construtores” na página 1889](#)
- [“Métodos de objeto \(público\)” na página 1889](#)
- [“Métodos de objeto \(protegidos\)” na página 1889](#)
- [“Códigos de Razão” na página 1889](#)

## Atributos do Objeto

### código de conclusão

O código de conclusão mais recente. O valor inicial é zero. Os seguintes valores adicionais são possíveis:

- MQCC\_OK
- MQCC\_WARNING
- MQCC\_FAILED

### **código de razão**

O código de razão mais recente O valor inicial é zero.

## **Construtores**

### **ImqError();**

O construtor padrão..

### **ImqError( const ImqError & error );**

O construtor de cópia..

## **Métodos de objeto (público)**

### **void operator = ( const ImqError & error );**

Copia os dados da instância de *erro*, substituindo os dados da instância existente

### **void clearErrorCódigos ();**

Configura o **código de conclusão** e o **código de razão** para zero.

### **MQLONG completionCode () const ;**

Retorna o **código de conclusão**.

### **MQLONG reasonCode () const ;**

Retorna o **código de razão**.

## **Métodos de objeto (protegidos)**

### **ImqBoolean checkReadPointer ( const void \* pointer, const size\_t length );**

Verifica se a combinação do ponteiro e do comprimento é válida para acesso somente leitura e retorna TRUE se for bem-sucedida

### **ImqBoolean checkWritePointer ( const void \* pointer, const size\_t length );**

Verifica se a combinação de ponteiro e comprimento é válida para acesso de leitura/gravação e retorna TRUE se for bem-sucedida

### **void setCompletionCode ( const MQLONG code = 0);**

Configura o **código de conclusão**

### **void setReasonCode ( const MQLONG code = 0);**

Configura o **código de razão**

## **Códigos de Razão**

- MQRC\_BUFFER\_ERROR

## **Classe C++ ImqGetMessageOptions**

Esta classe encapsula a estrutura de dados MQGMO

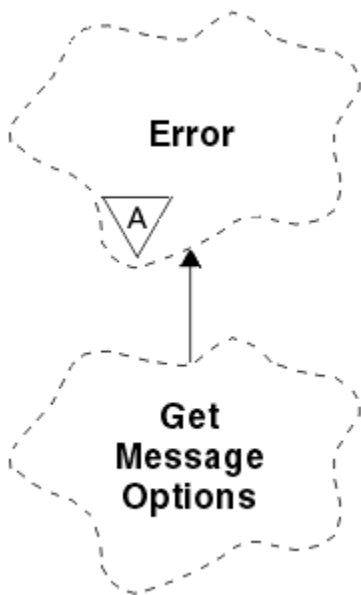


Figura 22. Classe *ImqGetMessageOptions*

- [“Atributos do Objeto”](#) na página 1890
- [“Construtores”](#) na página 1891
- [“Métodos de objeto \(público\)”](#) na página 1892
- [“Métodos de objeto \(protegidos\)”](#) na página 1893
- [“Dados do objeto \(protegidos\).”](#) na página 1893
- [“Códigos de Razão”](#) na página 1893

## Atributos do Objeto

### status do grupo

O status de uma mensagem para um grupo de mensagens. O valor inicial é MQGS\_NOT\_IN\_GROUP. Os seguintes valores adicionais são possíveis:

- MQGS\_MSG\_IN\_GROUP
- MQGS\_LAST\_MSG\_IN\_GROUP

### opções de correspondência

Opções para selecionar mensagens recebidas. O valor inicial é MQMO\_MATCH\_MSG\_ID | MQMO\_MATCH\_CORREL\_ID. Os seguintes valores adicionais são possíveis:

- MQMO\_GROUP\_ID
- MQMO\_MATCH\_MSG\_SEQ\_NUMBER
- MQMO\_MATCH\_OFFSET
- MQMO\_MSG\_TOKEN
- MQMO\_NONE

### message token (token de mensagem)

Token de mensagem Um valor binário (MQBYTE16) de comprimento MQ\_MSG\_TOKEN\_LENGTH. O valor inicial é MQMTOK\_NONE.

### opções

Opções aplicáveis a uma mensagem. O valor inicial é MQGMO\_NO\_WAIT. Os seguintes valores adicionais são possíveis:

- MQGMO\_WAIT
- MQGMO\_SYNCPOINT

- MQGMO\_SYNCPOINT\_IF\_PERSISTENT
- MQGMO\_NO\_SYNCPOINT
- MQGMO\_MARK\_SKIP\_BACKOUT
- MQGMO\_BROWSE\_FIRST
- MQGMO\_BROWSE\_NEXT
- MQGMO\_BROWSE\_MSG\_UNDER\_CURSOR
- MQGMO\_MSG\_UNDER\_CURSOR
- MQGMO\_LOCK
- MQGMO\_UNLOCK
- MQGMO\_ACCEPT\_TRUNCATED\_MSG
- MQGMO\_SET\_SIGNAL
- MQGMO\_FAIL\_IF QUIESCING
- MQGMO\_CONVERT
- MQGMO\_LOGICAL\_ORDER
- MQGMO\_COMPLETE\_MSG
- MQGMO\_ALL\_MSGS\_AVAILABLE
- MQGMO\_ALL\_SEGMENTS\_AVAILABLE
- MQGMO\_NONE

#### **Nome da fila resolvida**

Nome da fila resolvido. Esse atributo é somente leitura. Os nomes nunca têm mais de 48 caracteres e podem ser preenchidos com nulos. O valor inicial é uma cadeia nula.

#### **Comprimento retornado**

Comprimento retornado. O valor inicial é MQRL\_UNDEFINED. Esse atributo é somente leitura.

#### **segmentação**

A capacidade de segmentar uma mensagem O valor inicial é MQSEG\_INIBITED O valor adicional, MQSEG\_ALLOWED, é possível.

#### **status do segmento**

O status de segmentação de uma mensagem O valor inicial é MQSS\_NOT\_A\_SEGMENT. Os seguintes valores adicionais são possíveis:

- MQSS\_SEGMENT
- MQSS\_LAST\_SEGMENT

#### **participação de ponto de sincronização**

TRUE quando as mensagens são recuperadas sob o controle de ponto de sincronização

#### **intervalo de espera**

O período de tempo em que o método get da classe pausa enquanto aguarda uma mensagem adequada chegar, se uma ainda não estiver disponível. O valor inicial é zero, o que afeta uma espera indefinida. O valor adicional, MQWI\_UNLIMITED, é possível. Esse atributo é ignorado a menos que as opções incluam MQGMO\_WAIT.

## **Construtores**

#### **ImqGetMessageOptions( );**

O construtor padrão..

#### **ImqGetMessageOptions(const ImqGetMessageOptions & gmo );**

O construtor de cópia..

## Métodos de objeto (público)

**void operator = (const ImqGetMessageOptions & gmo );**

Copia os dados de instância do *gmo*, substituindo os dados de instância existentes

**MQCHAR groupStatus () const;**

Retorna o status do grupo..

**void setGroupStatus (const MQCHAR status );**

Configura o status do grupo

**MQLONG matchOptions () const;**

Retorna as opções de correspondência..

**void setMatchOptions (const MQLONG opções );**

Configura as opções de correspondência

**ImqBinary messageToken() const;**

Retorna o token da mensagem..

**ImqBoolean setMessageToken (const ImqBinary & );**

Configura o token da mensagem O comprimento de dados do *token* deve ser zero ou MQ\_MSG\_TOKEN\_LENGTH. Esse método retornará TRUE se for bem-sucedido

**void setMessageToken (const MQBYTE16 token = 0);**

Configura o token da mensagem *token* pode ser zero, que é o mesmo que especificar MQMTOK\_NONE. Se *token* for diferente de zero, ele deverá endereçar MQ\_MSG\_TOKEN\_LENGTH bytes de dados binários.

Ao usar valores predefinidos, como MQMTOK\_NONE, talvez não seja necessário fazer uma conversão para assegurar uma correspondência de assinatura, por exemplo (MQBYTE \*) MQMTOK\_NONE

**Opções MQLONG () const;**

Retorna as opções..

**void setOptions (const MQLONG options );**

Configura as opções, incluindo o valor de participação do ponto de sincronização

**ImqString resolvedQueueNome () const;**

Retorna uma cópia do nome da fila resolvida

**MQLONG returnedLength() const;**

Retorna o comprimento retornado..

**Segmentação MQCHAR () const;**

Retorna a segmentação..

**void setSegmentation (const MQCHAR value );**

Configura a segmentação

**MQCHAR segmentStatus () const;**

Retorna o status do segmento..

**void setSegmentStatus (const MQCHAR status );**

Define o status do segmento.

**ImqBoolean syncPointParticipação () const;**

Retorna o valor de participação do ponto de sincronização, que é TRUE se as opções incluírem MQGMO\_SYNCPOINT ou MQGMO\_SYNCPOINT\_IF\_PERSISTENT.

**void setSyncPointParticipation (const ImqBoolean sync );**

Configura o valor de participação do ponto de sincronização. Se *sync* for TRUE, alterará as opções para incluir MQGMO\_SYNCPOINT e excluir MQGMO\_NO\_SYNCPOINT e MQGMO\_SYNCPOINT\_IF\_PERSISTENT. Se *sync* for FALSE, altera as opções para incluir MQGMO\_NO\_SYNCPOINT e para excluir MQGMO\_SYNCPOINT e MQGMO\_SYNCPOINT\_IF\_PERSISTENT.

**MQLONG waitInterval () const;**

Retorna o intervalo de espera..



**void setWaitInterval (const MQLONG interval );**

Configura o intervalo de espera..

### Métodos de objeto (protegidos)

**static void setVersionSuportado (const MQLONG);**

Configura a versão do MQGMO O padrão é MQGMO\_VERSION\_3.

### Dados do objeto (protegidos).

#### **MQGMO omqgmo**

Uma estrutura de dados MQGMO Versão 2. Acesse apenas os campos MQGMO suportados para MQGMO\_VERSION\_2 .

#### **PMQGMO opgmo**

O endereço de uma estrutura de dados MQGMO.. O número da versão para esse endereço é indicado em *olVersion* Inspeccione o número da versão antes de acessar os campos MQGMO, para assegurar que eles estejam presentes

#### **MQLONG olVersion**

O número da versão da estrutura de dados MQGMO endereçado por *opgmo*..

### Códigos de Razão

- MQRC\_BINARY\_DATA\_LENGTH\_ERROR

## Classe C++ ImqHeader

Essa classe abstrata contém recursos comuns da estrutura de dados MQDLH.

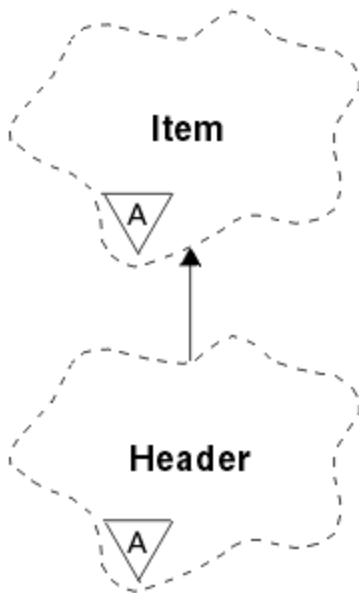


Figura 23. Classe ImqHeader

- [“Atributos do Objeto”](#) na página 1893
- [“Construtores”](#) na página 1894
- [“Métodos de objeto \(público\)”](#) na página 1894

### Atributos do Objeto

#### **conjunto de caracteres**

O identificador do conjunto de caracteres codificado original. Inicialmente MQCCSI\_Q\_MGR.

## **codificação**

A codificação original.. MQENC\_NATIVE inicialmente.

## **format**

O formato original. MQFMT\_NONE inicial.

## **signalizações de cabeçalho**

Os valores iniciais são:

- Zero para objetos da classe ImqDeadLetterHeader
- MQIIH\_NONE para objetos da classe ImqIMSBridgeHeader
- MQRMHF\_LAST para objetos da classe de cabeçalho ImqReference
- MQCIH\_NONE para objetos da classe ImqCICSBridgeHeader
- MQWIH\_NONE para objetos da classe de cabeçalho ImqWork

## **Construtores**

### **ImqHeader();**

O construtor padrão..

### **ImqHeader( const ImqHeader & header );**

O construtor de cópia..

## **Métodos de objeto (público)**

### **void operator = ( const ImqHeader & header );**

Copia os dados de instância do *header*, substituindo os dados de instância existentes

### **virtual MQLONG characterSet () const ;**

Retorna o **conjunto de caracteres**

### **virtual void setCharacterSet ( const MQLONG ccsid = MQCCSI\_Q\_MGR);**

Configura o **conjunto de caracteres**..

### **virtual MQLONG codificação () const ;**

Retorna a **codificação**..

### **virtual void setEncoding ( const MQLONG encoding = MQENC\_NATIVE);**

Configura a **de codificação**

### **virtual ImqString formato () const ;**

Retorna uma cópia do **formato**, incluindo espaços em branco à direita.

### **virtual void setFormat ( const char \* name = 0);**

Configura o **formato**, preenchido com 8 caracteres com espaços em branco à direita..

### **virtual MQLONG headerFlags () const ;**

Retorna os **signalizadores de cabeçalho**

### **virtual void setHeaderFlags ( const MQLONG flags = 0);**

Configura os **signalizadores de cabeçalho**

## **Classe C++ ImqIMSBridgeHeader**

Essa classe contém recursos da estrutura de dados MQIIH.

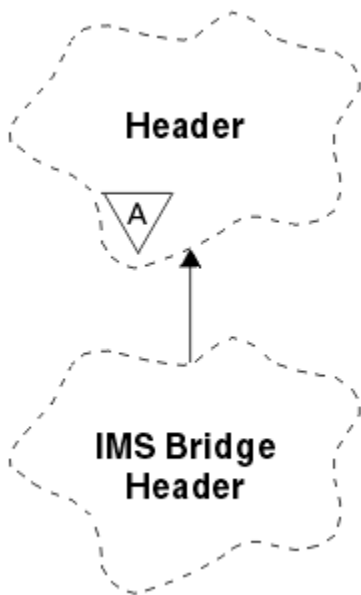


Figura 24. Classe *ImqIMSBridgeHeader*

Objetos dessa classe são usados por aplicativos que enviam mensagens para a ponte do IMS por IBM MQ for z/OS.

**Nota:** O conjunto de caracteres e a codificação *ImqHeader* devem ter valores padrão e não devem ser configurados para nenhum outro valor

- [“Atributos do Objeto”](#) na página 1895
- [“Construtores”](#) na página 1896
- [“Métodos \*ImqItem\* sobrecarregados”](#) na página 1896
- [“Métodos de objeto \(público\)”](#) na página 1896
- [“Dados do objeto \(protegidos\).”](#) na página 1897
- [“Códigos de Razão”](#) na página 1897

## Atributos do Objeto

### autenticador

Senha ou passticket do RACF , de comprimento `MQ_AUTHENTICATOR_LENGTH` O valor inicial é `MQIAUT_NONE`.

### Modo Comprometer

Modo de confirmação... Consulte o *Guia do Usuário do OTMA* para obter mais informações sobre IMS modos de confirmação. O valor inicial é `MQICM_COMMIT_THEN_SEND`. O valor adicional, `MQICM_SEND_THEN_COMMIT`, é possível.

### Intervenção terminal lógica

Substituição do terminal lógico, de comprimento `MQ_LTERM_OVERRIDE_LENGTH` O valor inicial é uma cadeia nula.

### Nome de mapa dos Serviços de Formato de Mensagens

nome do mapa MFS, de comprimento `MQ_MFS_MAP_NAME_LENGTH`. O valor inicial é uma cadeia nula.

### formato de resposta para

Formato de qualquer resposta de comprimento `MQ_FORMAT_LENGTH`. O valor inicial é `MQFMT_NONE`.

### escopo de segurança

Escopo do processamento de segurança do IMS O valor inicial é `MQISS_CHECK`. O valor adicional, `MQISS_FULL`, é possível.

### **id da instância de transação**

Identidade da instância de transação, um valor binário (MQBYTE16) de comprimento MQ\_TRAN\_INSTANCE\_ID\_LENGTH. O valor inicial é MQITII\_NONE.

### **ESTADO DA TRANSAÇÃO**

Estado da conversa IMS . O valor inicial é MQITS\_NOT\_IN\_CONVERSATION. O valor adicional, MQITS\_IN\_CONVERSATION, é possível.

## **Construtores**

### **ImqIMSBridgeHeader();**

O construtor padrão..

### **ImqIMSBridgeHeader(const ImqIMSBridgeHeader & );**

O construtor de cópia..

## **Métodos ImqItem sobrecarregados**

### **virtual ImqBoolean copyOut ( ImqMessage & msg );**

Insere uma estrutura de dados MQIIH no buffer de mensagem no início, movendo os dados da mensagem existentes mais adiante Configura o formato de *msg* para MQFMT\_IMS

Consulte a descrição do método da classe-pai para obter detalhes adicionais

### **virtual ImqBoolean pasteIn ( ImqMessage & msg );**

Lê uma estrutura de dados MQIIH do buffer de mensagem.

Para ser bem sucedido, a codificação do objeto *msg* deve ser MQENC\_NATIVE Recuperar mensagens com MQGMO\_CONVERT para MQENC\_NATIVE.

Para ser bem-sucedido, o formato ImqMessage deve ser MQFMT\_IMS..

Consulte a descrição do método da classe-pai para obter detalhes adicionais

## **Métodos de objeto (público)**

### **operador void = (const ImqIMSBridgeHeader & header );**

Copia os dados de instância do *header*, substituindo os dados de instância existentes

### **Autenticador ImqString () const;**

Retorna uma cópia do autenticador, preenchidos com espaços em branco à direita para o comprimento MQ\_AUTHENTICATOR\_LENGTH

### **void setAuthenticator (const char \* name );**

Configura o autenticador

### **MQCHAR commitMode () const;**

Retorna o modo de confirmação

### **void setCommitMode (const MQCHAR mode );**

Configura o modo de confirmação

### **ImqString logicalTerminalSubstituir () const;**

Retorna uma cópia da substituição do terminal lógico

### **void setLogicalTerminalOverride (const char \* override );**

Configura a substituição do Terminal Lógico.

### **ImqString messageFormatServicesMapNome () const;**

Retorna uma cópia do nome do mapa de serviços de formato de mensagem

### **void setMessageFormatServicesMapName (const char \* name );**

Configura o nome de mapa de serviços de formato de mensagem

### **ImqString replyToFormat () const;**

Retorna uma cópia do formato de resposta, preenchido com espaços em branco à direita para comprimento MQ\_FORMAT\_LENGTH.

**void setReplyToFormat (const char \* *format* );**

Configura o formato de resposta, preenchido com espaços em branco à direita para o comprimento MQ\_FORMAT\_LENGTH

**MQCHAR securityScope () const;**

Retorna o escopo de segurança..

**void setSecurityScope (const MQCHAR *scope* );**

Configura o escopo de segurança

**ImqBinary transactionInstanceId () const;**

Retorna uma cópia do ID da instância de transação

**ImqBoolean setTransactionInstanceId (const ImqBinary & *id* );**

Configura o ID da instância de transação O comprimento de dados do *token* deve ser zero ou MQ\_TRAN\_INSTANCE\_ID\_LENGTH. Esse método retornará TRUE se for bem-sucedido

**void setTransactionInstanceId (const MQBYTE16 *id* = 0);**

Configura o ID da instância de transação *id* pode ser zero, que é o mesmo que especificar MQITII\_NONE. Se *id* for diferente de zero, ele deverá abordar MQ\_TRAN\_INSTANCE\_ID\_LENGTH bytes de dados binários. Ao usar valores predefinidos como MQITII\_NONE, você pode precisar fazer uma conversão para assegurar uma correspondência de assinatura, por exemplo (MQBYTE \*) MQITII\_NONE.

**MQCHAR transactionState () const;**

Retorna o estado da transação..

**void setTransactionState (const MQCHAR *state* );**

Configura o estado de transação

## Dados do objeto (protegidos).

**MQIIH omqiih**

A estrutura de dados do MQIIH

## Códigos de Razão

- MQRC\_BINARY\_DATA\_LENGTH\_ERROR
- MQRC\_INCONSISTENT\_FORMAT
- MQRC\_ENCODING\_ERROR
- MQRC\_STRUC\_ID\_ERROR

## Classe C++ ImqItem

Esta classe abstrata representa um item, talvez um de vários, dentro de uma mensagem.

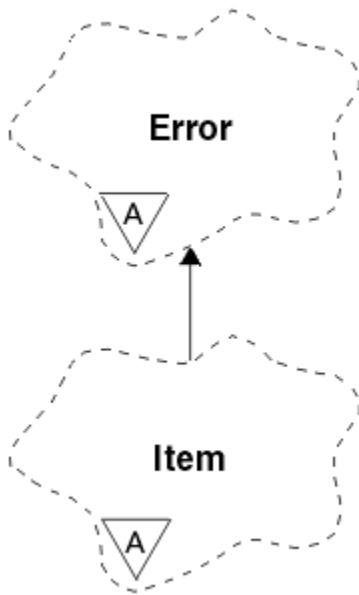


Figura 25. Classe *ImqItem*

Os itens são concatenados juntos em um buffer de mensagens. Cada especialização é associada a uma estrutura de dados específica que começa com um ID de estrutura...

Métodos polimórficos nesta classe abstrata permitem que itens sejam copiados para e a partir de mensagens. Os métodos *ImqMessage* **readItem** e **writeItem** fornecem outro estilo de chamar esses métodos polimórficos que são mais naturais para programas de aplicativos.

- [“Atributos do Objeto” na página 1898](#)
- [“Construtores” na página 1898](#)
- [“Métodos de classe \(público\).” na página 1898](#)
- [“Métodos de objeto \(público\)” na página 1899](#)
- [“Códigos de Razão” na página 1899](#)

## Atributos do Objeto

### id da estrutura

Uma sequência de quatro caracteres no início da estrutura de dados. Esse atributo é somente leitura. Considere esse atributo para as classes derivadas. Ele não é incluído automaticamente.

## Construtores

### **ImqItem();**

O construtor padrão..

### **ImqItem( const ImqItem & item );**

O construtor de cópia..

## Métodos de classe (público).

### **static ImqBoolean structureIds ( const char \* structure-id-to-test, const ImqMessage & msg );**

Retorna TRUE se o **ID da estrutura** do próximo *ImqItem* no *msg* recebido for igual ao *structure-id-to-test*. O próximo item é identificado como a parte do buffer de mensagem atualmente endereçada pelo *ImqCache* **ponteiro de dados**. Este método depende do **id da estrutura** e, portanto, não é garantido que funcione para todas as classes derivadas de *ImqItem*.

## Métodos de objeto (público)

### **void operator = ( const ImqItem & item );**

Copia os dados de instância do *item*, substituindo os dados de instância existentes

### **virtual ImqBoolean copyOut ( ImqMessage & msg ) = 0;**

Grava esse objeto como o próximo item em um buffer de mensagem de saída, anexando-o a quaisquer itens existentes. Se a operação de gravação for bem-sucedida, aumentará o ImqCache **comprimento de dados**. Esse método retornará TRUE se for bem-sucedido.

Substitua esse método para trabalhar com uma subclasse específica.

### **virtual ImqBoolean pasteIn ( ImqMessage & msg ) = 0;**

Lê esse objeto *destrutivamente* do buffer de mensagem recebida. A leitura é destrutiva em que o ImqCache **ponteiro de dados** é movido. No entanto, o conteúdo do buffer permanece o mesmo, portanto, os dados podem ser relidos reconfigurando o ponteiro de dados ImqCache.

A (sub) classe desse objeto deve ser consistente com o **ID da estrutura** localizado a seguir no buffer de mensagens do objeto *msg* ..

A **codificação** do objeto *msg* deve ser MQENC\_NATIVE.. É recomendado que as mensagens sejam recuperadas com a ImqMessage **encoding** configurada como MQENC\_NATIVE e com as ImqGetMessageOptions **opções**, incluindo MQGMO\_CONVERT.

Se a operação de leitura for bem-sucedida, o ImqCache **comprimento de dados** será reduzido. Esse método retornará TRUE se for bem-sucedido.

Substitua esse método para trabalhar com uma subclasse específica.

## Códigos de Razão

- MQRC\_ENCODING\_ERROR
- MQRC\_STRUC\_ID\_ERROR
- MQRC\_INCONSISTENT\_FORMAT
- MQRC\_INSUFENT\_BUFFER
- MQRC\_INSUFENT\_DATA

## Classe C++ ImqMessage

Essa classe encapsula uma estrutura de dados MQMD e também manipula a construção e a reconstrução de dados da mensagem.

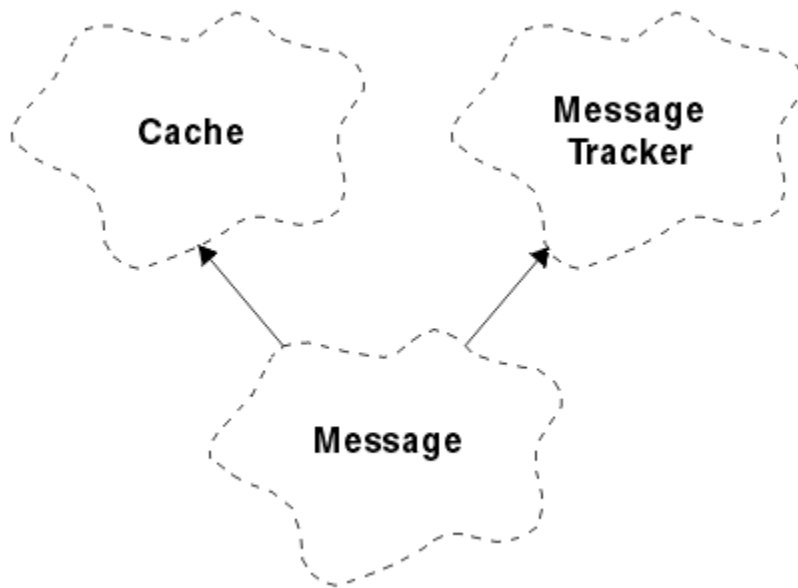


Figura 26. Classe *ImqMessage*

- [“Atributos do Objeto”](#) na página 1900
- [“Construtores”](#) na página 1904
- [“Métodos de objeto \(público\)”](#) na página 1904
- [“Métodos de objeto \(protegidos\)”](#) na página 1906
- [“Dados do objeto \(protegidos\).”](#) na página 1906

## Atributos do Objeto

### Dados do ID do aplicativo

Informações de identidade associadas a uma mensagem.. O valor inicial é uma cadeia nula.

### Dados de origem do aplicativo

As informações de origem associadas a uma mensagem O valor inicial é uma cadeia nula.

### contagem de restauração

O número de vezes que uma mensagem foi recuperada provisoriamente e subsequentemente restaurada. O valor inicial é zero. Esse atributo é somente leitura.

### conjunto de caracteres

ID do conjunto de caracteres codificados. O valor inicial é MQCCSI\_Q\_MGR.. Os seguintes valores adicionais são possíveis:

- MQCCSI\_INHERIT
- MQCCSI\_EMBEDDED

Também é possível usar um ID do conjunto de caracteres codificados de sua escolha. Para obter informações sobre isso, consulte [“Conversão de página de códigos”](#) na página 971.

### codificação

A codificação da máquina dos dados da mensagem O valor inicial é MQENC\_NATIVE.

### expiração

Uma quantidade dependente de tempo que controla por quanto tempo o IBM MQ retém uma mensagem não recuperada antes de descartá-la, O valor inicial é MQEI\_UNLIMITED.

### format

O nome do formato (modelo) que descreve o layout de dados no buffer. Nomes maiores que oito caracteres são truncados para oito caracteres. Nomes são sempre preenchidos com espaços em branco para oito caracteres. O valor constante inicial é MQFMT\_NONE. As seguintes constantes adicionais são possíveis:



- MQFMT\_ADMIN
- MQFMT\_CICS
- MQFMT\_COMMAND\_1
- MQFMT\_COMMAND\_2
- MQFMT\_DEAD\_LETTER\_HEADER
- MQFMT\_DIST\_HEADER
- MQFMT\_EVENT
- MQFMT\_IMS
- MQFMT\_IMS\_VAR\_STRING
- MQFMT\_MD\_EXTENSION
- MQFMT\_PCF
- MQFMT\_REF\_MSG\_HEADER
- MQFMT\_RF\_HEADER
- MQFMT\_STRING
- MQFMT\_TRIGGER
- MQFMT\_WORK\_INFO\_HEADER
- MQFMT\_XMIT\_Q\_HEADER

Também é possível usar uma sequência específica do aplicativo de sua escolha. Para obter mais informações sobre isso, consulte o campo [“Formato \(MQCHAR8\) para MQMD”](#) na página 462 do descritor de mensagens (MQMD)..

#### **Sinalizações de mensagem**

Informações de controle de segmentação O valor inicial é MQMF\_SEGMENTATION\_IBITED. Os seguintes valores adicionais são possíveis:

- MQMF\_SEGMENTATION\_ALLOWED
- MQMF\_MSG\_IN\_GROUP
- MQMF\_LAST\_MSG\_IN\_GROUP
- MQMF\_SEGMENT
- MQMF\_LAST\_SEGMENT
- MQMF\_NONE

#### **tipo de mensagem**

A categorização ampla de uma mensagem O valor inicial é MQMT\_DATAGRAM. Os seguintes valores adicionais são possíveis:

- MQMT\_SYSTEM\_FIRST
- MQMT\_SYSTEM\_LAST
- MQMT\_DATAGRAM
- MQMT\_REQUEST
- MQMT\_REPLY
- MQMT\_REPORT
- MQMT\_APPL\_FIRST
- MQMT\_APPL\_LAST

Também é possível usar um valor específico do aplicativo de sua escolha.. Para obter mais informações sobre isso, consulte o campo [“MsgType \(MQLONG\) para MQMD”](#) na página 451 do descritor de mensagens (MQMD)..

#### **deslocamento**

Informações de deslocamento O valor inicial é zero.

### **Comprimento original**

O comprimento original de uma mensagem segmentada. O valor inicial é MQOL\_UNDEFINED.

### **persistence**

Indica que a mensagem é importante e deve ser sempre submetida a backup usando armazenamento persistente. Essa opção implica uma multa de desempenho. O valor inicial é MQPER\_PERSISTENCE\_AS\_Q\_DEF. Os seguintes valores adicionais são possíveis:

- MQPER\_PERSISTENT
- MQPER\_NOT\_PERSISTENT

### **priority**

A prioridade relativa para transmissão e entrega. Mensagens da mesma prioridade geralmente são entregues na mesma sequência em que foram fornecidas (embora haja vários critérios que devem ser atendidos para garantir isso). O valor inicial é MQPRI\_PRIORITY\_AS\_Q\_DEF.

### **Validação da propriedade**

Especifica se a validação de propriedades deve ocorrer quando uma propriedade da mensagem é configurada O valor inicial é **MQCMHO\_DEFAULT\_VALIDATION**. Os seguintes valores adicionais são possíveis:

- MQCMHO\_VALIDATE
- MQCMHO\_NO\_VALIDATION

Os métodos a seguir agem na **validação de propriedades**:

#### **MQQLONG propertyValidation() const;**

Retorna a opção **validação de propriedade**

#### **void setPropertyValidação (const MQQLONG opção );**

Configura a opção de **validação de propriedade**

### **Colocar nome do aplicativo**

O nome do aplicativo que colocou uma mensagem O valor inicial é uma cadeia nula.

### **Colocar tipo de aplicativo**

O tipo de aplicativo que coloca uma mensagem. O valor inicial é MQAT\_NO\_CONTEXT. Os seguintes valores adicionais são possíveis:

- MQAT\_AIX
- MQAT\_CICS
- MQAT\_CICS\_BRIDGE
- MQAT\_DOS
- MQAT\_IMS
- MQAT\_IMS\_BRIDGE
- MQAT\_MVS
- MQAT\_NOTES\_AGENT
- MQAT\_OS2
- MQAT\_OS390
- MQAT\_OS400
- MQAT\_QMGR
- MQAT\_UNIX
- MQAT\_WINDOWS
- MQAT\_WINDOWS\_NT
- MQAT\_XCF
- MQAT\_DEFAULT
- MQAT\_UNKNOWN

- MQAT\_USER\_FIRST
- MQAT\_USER\_LAST

Também é possível usar uma sequência específica do aplicativo de sua escolha Para obter mais informações sobre isso, consulte o campo [“PutApplTipo \(MQLONG\) para MQMD”](#) na página 477 do descritor de mensagens (MQMD)..

#### **Data de colocação**

A data na qual uma mensagem foi colocada O valor inicial é uma cadeia nula.

#### **Horário de colocação**

O horário em que uma mensagem foi colocada O valor inicial é uma cadeia nula.

#### **Nome do Gerenciador de Filas de Resposta**

O nome do gerenciador de filas para o qual qualquer resposta deve ser enviada O valor inicial é uma cadeia nula.

#### **Nome da Fila de Resposta**

O nome da fila para a qual qualquer resposta deve ser enviada O valor inicial é uma cadeia nula.

#### **relatório**

Informações de feedback associadas a uma mensagem.. O valor inicial é MQRO\_NONE. Os seguintes valores adicionais são possíveis:

- MQRO\_EXCEPTION
- MQRO\_EXCEPTION\_WITH\_DATA
- MQRO\_EXCEPTION\_WITH\_FULL\_DATA \*
- MQRO\_EXPIRATION
- MQRO\_EXPIRATION\_WITH\_DATA
- MQRO\_EXPIRATION\_WITH\_FULL\_DATA \*
- MQRO\_COA
- MQRO\_COA\_WITH\_DATA
- MQRO\_COA\_WITH\_FULL\_DATA \*
- MQRO\_COD
- MQRO\_COD\_WITH\_DATA
- MQRO\_COD\_WITH\_FULL\_DATA \*
- MQRO\_PAN
- MQRO\_NAN
- MQRO\_NEW\_MSG\_ID
- MQRO\_NEW\_CORREL\_ID
- MQRO\_COPY\_MSG\_ID\_TO\_CORREL\_ID
- MQRO\_PASS\_CORREL\_ID
- MQRO\_DEAD\_LETTER\_Q
- MQRO\_DISCARD\_MSG

em que \* indica valores que não são suportados no IBM MQ for z/OS.

#### **número de sequência**

Informações de sequência que identificam uma mensagem em um grupo. O valor inicial é um.

#### **comprimento total da mensagem**

O número de bytes que estavam disponíveis durante a tentativa mais recente de leitura de uma mensagem Esse número será maior que o ImqCache **comprimento da mensagem** se a última mensagem foi truncada ou se a última mensagem não foi lida porque o truncamento teria ocorrido. Esse atributo é somente leitura. O valor inicial é zero.

Esse atributo pode ser útil em qualquer situação que envolva mensagens truncadas

## ID do usuário

Uma identidade do usuário associada a uma mensagem O valor inicial é uma cadeia nula.

## Construtores

### **ImqMessage( );**

O construtor padrão..

### **ImqMessage( const ImqMessage & msg );**

O construtor de cópia.. Consulte o método **operator =** para obter detalhes.

## Métodos de objeto (público)

### **void operator = ( const ImqMessage & msg );**

Copia o MQMD e os dados da mensagem de *msg*. Se um buffer tiver sido fornecido pelo usuário para esse objeto, a quantidade de dados copiados será restrita ao tamanho do buffer disponível. Caso contrário, o sistema assegura que um buffer de tamanho adequado seja disponibilizado para os dados copiados..

### **ImqString applicationIdDados () const ;**

Retorna uma cópia dos **dados do ID do aplicativo**

### **void setApplicationIdData ( const char \* dados = 0 );**

Configura os **dados do ID do aplicativo**

### **ImqString applicationOriginDados () const ;**

Retorna uma cópia dos **dados de origem do aplicativo**

### **void setApplicationOriginData ( const char \* data = 0 );**

Configura os **dados de origem do aplicativo**

### **MQLONG backoutCount () const ;**

Retorna a **contagem de restauração**.

### **MQLONG characterSet () const ;**

Retorna o **conjunto de caracteres**

### **void setCharacterSet ( const MQLONG ccsid = MQCCSI\_Q\_MGR );**

Configura o **conjunto de caracteres**..

### **MQLONG codificação () const ;**

Retorna a **codificação**..

### **void setEncoding ( const MQLONG encoding = MQENC\_NATIVE );**

Configura a **de codificação**

### **MQLONG expiração () const ;**

Retorna **expiração**.

### **void setExpiry ( const MQLONG expire );**

Configura o **de expiração**

### **ImqString formato () const ;**

Retorna uma cópia do **formato**, incluindo espaços em branco à direita.

### **ImqBoolean formatIs ( const char \* format-to-test ) const ;**

Retornará TRUE se o **format** for o mesmo que *format-to-test*.

### **void setFormat ( const char \* name = 0 );**

Configura o **formato**, preenchido com oito caracteres com espaços em branco à direita.

### **MQLONG messageFlags () const ;**

Retorna os **sinalizadores de mensagem**

### **void setMessageFlags ( const MQLONG flags );**

Configura os **sinalizadores de mensagem**

### **MQLONG messageType () const ;**

Retorna o **tipo de mensagem**..

**anular setMessageTipo ( const MQLONG *tipo* )**  
 Configura o **tipo de mensagem**..

**MQLONG offset () const ;**  
 Retorna o **offset**.

**void setOffset ( const MQLONG *offset* );**  
 Configura o **offset**.

**MQLONG originalLength () const ;**  
 Retorna o **comprimento original**..

**void setOriginalLength ( const MQLONG *length* );**  
 Configura o **comprimento original**

**MQLONG persistência () const ;**  
 Retorna a **persistência**..

**void setPersistence ( const MQLONG *persistência* ); e**  
 Configura a **persistência**..

**MQLONG priority () const ;**  
 Retorna a **prioridade**

**void setPriority ( const MQLONG *priority* );**  
 Configura a **prioridade**

**ImqString putApplicationNome () const ;**  
 Retorna uma cópia do **nome do aplicativo put**..

**void setPutApplicationName ( const char \* *name* = 0);**  
 Configura o **nome do aplicativo de colocação**

**MQLONG putApplicationTipo () const ;**  
 Retorna o **tipo de aplicativo put**..

**void setPutApplicationType ( const MQLONG *type* = MQAT\_NO\_CONTEXT);**  
 Configura o **tipo de aplicativo put**

**ImqString putDate () const ;**  
 Retorna uma cópia da **data de colocação**

**void setPutDate ( const char \* *date* = 0);**  
 Configura a **data de colocação**

**ImqString putTime () const ;**  
 Retorna uma cópia do **horário de colocação**

**void setPutTime ( const char \* *time* = 0);**  
 Configura o **tempo de colocação**

**ImqBoolean readItem ( ImqItem & *item* );**  
 Lê no objeto *item* do buffer de mensagem, usando o método ImqItem **pasteIn** . Ele retornará TRUE se for bem-sucedido

**ImqString replyToQueueManagerNome () const ;**  
 Retorna uma cópia do **nome do gerenciador de filas de resposta**..

**void setReplyToQueueManagerName ( const char \* *nome* = 0);**  
 Configura o **nome do gerenciador de filas de resposta**

**ImqString replyToQueueName () const ;**  
 Retorna uma cópia do **reply-to nome da fila**.

**void setReplyToQueueName ( const char \* *name* = 0);**  
 Configura o **nome da fila de resposta**..

**MQLONG relatório () const ;**  
 Retorna o **relatório**..

**void setReport ( const MQLONG *report* );**  
 Configura o **relatório**

**MQLONG sequenceNumber () const ;**

Retorna o **número de sequência**

**void setSequenceNumber ( const MQLONG number );**

Configura o **número de sequência**

**size\_t totalMessageComprimento () const ;**

Retorna o **comprimento total da mensagem..**

**ImqString userId () const ;**

Retorna uma cópia do **ID do usuário**

**void setUserId ( const char \* id = 0);**

Configura o **ID do usuário**

**ImqBoolean writeItem (item ImqItem &);**

Grava do objeto *item* no buffer de mensagem, usando o método ImqItem **copyOut** . A gravação pode assumir o formato de inserção, substituição ou anexação: isso depende da classe do objeto *item* .

Esse método retornará TRUE se for bem-sucedido

## Métodos de objeto (protegidos)

**vazio estático setVersionSuportado ( const MQLONG )**

Configura a **versão do MQMD** O padrão é **MQMD\_VERSION\_2**.

## Dados do objeto (protegidos).

**z/OS** **MQMD1 omqmd**

A estrutura de dados do MQMD no z/OS

**Multi** **MQMD2 omqmd**

A estrutura de dados do MQMD no [Multiplataformas](#)

## Classe C++ do Rastreador ImqMessage

Essa classe encapsula esses atributos de um objeto ImqMessage ou ImqQueue que pode estar associado a qualquer objeto.

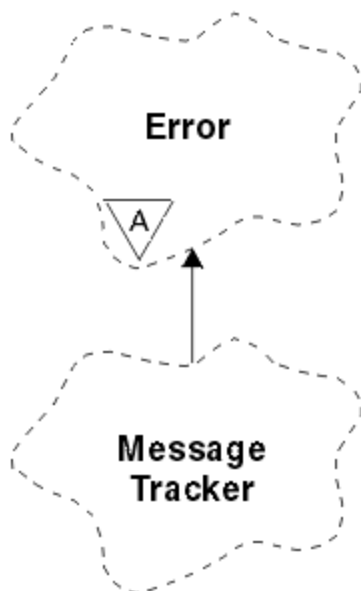


Figura 27. Classe do Rastreador ImqMessage

Essa classe está relacionada às chamadas MQI listadas em [“Referência cruzada do ImqMessageTracker”](#) na página 1855.

- [“Atributos do Objeto” na página 1907](#)
- [“Construtores” na página 1908](#)
- [“Métodos de objeto \(público\)” na página 1908](#)
- [“Códigos de Razão” na página 1909](#)

## Atributos do Objeto

### token de conta

Um valor binário (MQBYTE32) de comprimento MQ\_ACCOUNTING\_TOKEN\_LENGTH. O valor inicial é MQACT\_NONE.

### ID de correlação

Um valor binário (MQBYTE24) de comprimento MQ\_CORREL\_ID\_LENGTH que você designa para mensagens correlacionadas. O valor inicial é MQCI\_NONE. O valor adicional, MQCI\_NEW\_SESSION, é possível.

### feedback

As informações de feedback a serem enviadas com uma mensagem. O valor inicial é MQFB\_NONE. Os seguintes valores adicionais são possíveis:

- MQFB\_SYSTEM\_FIRST
- MQFB\_SYSTEM\_LAST
- MQFB\_APPL\_FIRST
- MQFB\_APPL\_LAST
- MQFB\_COA
- MQFB\_COD
- MQFB\_EXPIRATION
- MQFB\_PAN
- MQFB\_NAN
- MQFB\_QUIT
- MQFB\_DATA\_LENGTH\_ZERO
- MQFB\_DATA\_LENGTH\_NEGATIVE
- MQFB\_DATA\_LENGTH\_TOO\_BIG
- MQFB\_BUFFER\_OVERFLOW
- MQFB\_LENGTH\_OFF\_BY\_ONE
- MQFB\_IIH\_ERROR
- MQFB\_NOT\_AUTHORIZED\_FOR\_IMS
- MQFB\_IMS\_ERROR
- MQFB\_IMS\_FIRST
- MQFB\_IMS\_LAST
- MQFB\_CICS\_APPL\_ABENDED
- MQFB\_CICS\_APPL\_NOT\_STARTED
- MQFB\_CICS\_BRIDGE\_FAILURE
- MQFB\_CICS\_CCSID\_ERROR
- MQFB\_CICS\_CIH\_ERROR
- MQFB\_CICS\_COMMAREA\_ERROR
- MQFB\_CICS\_CORREL\_ID\_ERROR
- MQFB\_CICS\_DLQ\_ERROR
- MQFB\_CICS\_ENCODING\_ERROR

- MQFB\_CICS\_INTERNAL\_ERROR
- MQFB\_CICS\_NOT\_AUTHORIZED
- MQFB\_CICS\_UOW\_BACKED\_OUT
- MQFB\_CICS\_UOW\_ERROR

Também é possível usar uma sequência específica do aplicativo de sua escolha Para obter mais informações sobre isso, consulte o campo [“Feedback \(MQLONG\) para MQMD” na página 456 do descritor de mensagens \(MQMD\)](#)..

### **ID de grupo**

Um valor binário (MQBYTE24) de comprimento MQ\_GROUP\_ID\_LENGTH exclusivo em uma fila. O valor inicial é MQGI\_NONE.

### **ID da mensagem**

Um valor binário (MQBYTE24) de comprimento MQ\_MSG\_ID\_LENGTH exclusivo em uma fila. O valor inicial é MQMI\_NONE.

## **Construtores**

### **ImqMessageTracker ();**

O construtor padrão..

### **ImqMessageTracker ( const ImqMessageTracker & tracker );**

O construtor de cópia.. Consulte o método **operator =** para obter detalhes.

## **Métodos de objeto (público)**

### **void operator = ( const ImqMessageTracker & tracker );**

Copia os dados de instância do *tracker*, substituindo os dados de instância existentes

### **ImqBinary accountingToken () const ;**

Retorna uma cópia do **token contábil**

### **ImqBoolean setAccountingToken ( const ImqBinary & token );**

Configura o **token de contabilidade**.. O **comprimento de dados** do *token* deve ser zero ou MQ\_ACCOUNTING\_TOKEN\_LENGTH. Esse método retornará TRUE se for bem-sucedido

### **void setAccountingToken ( const MQBYTE32 token = 0);**

Configura o **token de contabilidade**.. *token* pode ser zero, que é o mesmo que especificar MQACT\_NONE. Se *token* for diferente de zero, ele deverá abordar MQ\_ACCOUNTING\_TOKEN\_LENGTH bytes de dados binários. Ao usar valores predefinidos como MQACT\_NONE, pode ser necessário fazer uma conversão para assegurar uma correspondência de assinatura; por exemplo, (MQBYTE \*) MQACT\_NONE.

### **ImqBinary correlationId () const ;**

Retorna uma cópia do **correlation id**.

### **ImqBoolean setCorrelationId ( const ImqBinary & token );**

Configura o **ID de correlação**.. O **comprimento de dados** de *token* deve ser zero ou MQ\_CORREL\_ID\_LENGTH. Esse método retornará TRUE se for bem-sucedido

### **void setCorrelationId ( const MQBYTE24 id = 0);**

Configura o **ID de correlação**.. *id* pode ser zero, que é o mesmo que especificar MQCI\_NONE. Se *id* for diferente de zero, ele deverá endereçar MQ\_CORREL\_ID\_LENGTH bytes de dados binários. Ao usar valores predefinidos como MQCI\_NONE, talvez seja necessário fazer uma conversão para assegurar uma correspondência de assinatura; por exemplo, (MQBYTE \*) MQCI\_NONE.

### **MQLONG feedback () const ;**

Retorna o **feedback**..

### **void setFeedback ( const MQLONG feedback ); e**

Configura o **feedback**..

### **ImqBinary groupId () const ;**

Retorna uma cópia do **ID do grupo**



**ImqBoolean setGroupId ( const ImqBinary & token );**

Configura o **ID do grupo** O **comprimento de dados** do *token* deve ser zero ou MQ\_GROUP\_ID\_LENGTH. Esse método retornará TRUE se for bem-sucedido

**void setGroupId ( const MQBYTE24 id = 0);**

Configura o **ID do grupo** *id* pode ser zero, que é o mesmo que especificar MQGI\_NONE. Se *id* for diferente de zero, ele deverá endereçar bytes MQ\_GROUP\_ID\_LENGTH de dados binários. Ao usar valores predefinidos como MQGI\_NONE, pode ser necessário fazer uma conversão para assegurar uma correspondência de assinatura, por exemplo (MQBYTE \*) MQGI\_NONE.

**ImqBinary messageId () const ;**

Retorna uma cópia do **ID de mensagem**

**ImqBoolean setMessageId ( const ImqBinary & token );**

Configura o **ID de mensagem** O **comprimento de dados** do *token* deve ser zero ou MQ\_MSG\_ID\_LENGTH. Esse método retornará TRUE se for bem-sucedido

**void setMessageId ( const MQBYTE24 id = 0);**

Configura o **ID de mensagem** *id* pode ser zero, que é o mesmo que especificar MQMI\_NONE. Se *id* for diferente de zero, ele deverá endereçar bytes MQ\_MSG\_ID\_LENGTH de dados binários. Ao usar valores predefinidos como MQMI\_NONE, pode ser necessário fazer uma conversão para assegurar uma correspondência de assinatura, por exemplo (MQBYTE \*) MQMI\_NONE.

**Códigos de Razão**

- MQRC\_BINARY\_DATA\_LENGTH\_ERROR

**Classe C++ ImqNamelist**

Esta classe contém uma lista de nomes.

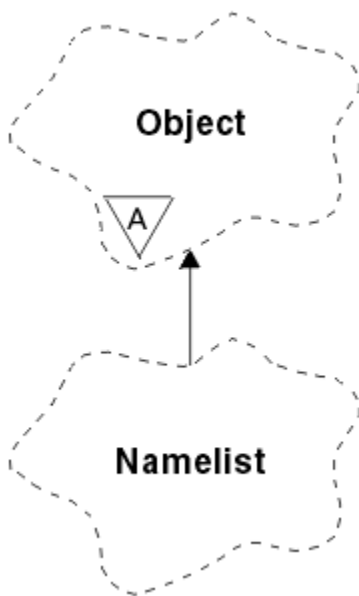


Figura 28. Classe ImqNamelist

Essa classe está relacionada às chamadas MQI listadas em [“Referência cruzada ImqNamelist”](#) na página 1855.

- [“Atributos do Objeto”](#) na página 1910
- [“Construtores”](#) na página 1910
- [“Métodos de objeto \(público\)”](#) na página 1910
- [“Códigos de Razão”](#) na página 1910

## Atributos do Objeto

### Contagem de nomes

O número de nomes de objetos em **nomes de lista de nomes**. Esse atributo é somente leitura.

### nomes da lista de nomes

Nomes de objetos, cujo número é indicado pela **contagem de nomes**. Esse atributo é somente leitura.

## Construtores

### **ImqNamelist();**

O construtor padrão..

### **ImqNamelist(const ImqNamelist & list);**

O construtor de cópia.. O ImqObject **status de abertura** é false

### **ImqNamelist(const char \* name );**

Configura o nome de ImqObject para **name**

## Métodos de objeto (público)

### **void operator = (const ImqNamelist e list );**

Copia os dados de instância da *lista*, substituindo os dados de instância existentes O ImqObject **status de abertura** é false

### **ImqBoolean nameCount(MQLONG & count );**

Fornecer uma cópia da **contagem de nomes** Ele retornará TRUE se for bem-sucedido

### **MQLONG nameCount ();**

Retorna a **contagem de nomes** sem qualquer indicação de possíveis erros

### **ImqBoolean namelistName (const MQLONG index, ImqString & name );**

Fornecer uma cópia de um dos **nomes de lista de nomes** por um índice baseado em zero Ele retornará TRUE se for bem-sucedido

### **ImqString namelistName (const MQLONG index );**

Retorna um dos **nomes de lista de nomes** por índice baseado em zero sem qualquer indicação de possíveis erros

## Códigos de Razão

- MQRC\_INDEX\_ERROR
- MQRC\_INDEX\_NOT\_PRESENT

## Classe C++ ImqObject

Esta classe é abstrata. Quando um objeto dessa classe é destruído, ele é automaticamente fechado e sua conexão do Gerenciador de ImqQueue é interrompida

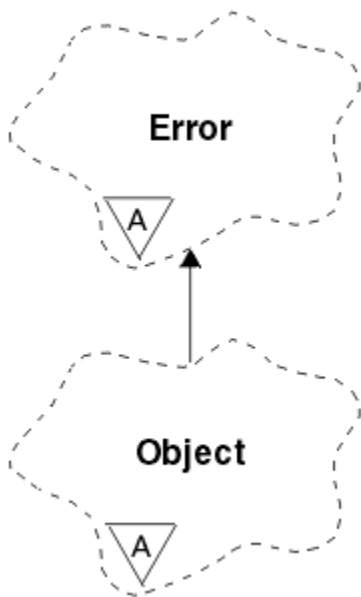


Figura 29. Classe *ImqObject*

Essa classe está relacionada às chamadas MQI listadas em [“Referência cruzada ImqObject”](#) na página 1855.

- [“Atributos de classe.”](#) na página 1911
- [“Atributos do Objeto”](#) na página 1911
- [“Construtores”](#) na página 1913
- [“Métodos de classe \(público\).”](#) na página 1913
- [“Métodos de objeto \(público\)”](#) na página 1913
- [“Métodos de objeto \(protegidos\)”](#) na página 1915
- [“Dados do objeto \(protegidos\).”](#) na página 1916
- [“Códigos de Razão”](#) na página 1916
- 

## Atributos de classe.

### comportamento

Controla o comportamento de abertura implícita.

#### **IMQ\_IMPL\_OPEN (8L)**

A abertura implícita é permitida Esse é o padrão.

## Atributos do Objeto

### Data da alteração

A data de alteração Esse atributo é somente leitura.

### Hora da alteração

O tempo de alteração Esse atributo é somente leitura.

### ID do usuário alternativo

O ID do usuário alternativo, até caracteres MQ\_USER\_ID\_LENGTH. O valor inicial é uma cadeia nula.

### ID de segurança alternativo

O ID de segurança alternativo. Um valor binário (MQBYTE40) de comprimento MQ\_SECURITY\_ID\_LENGTH. O valor inicial é MQSID\_NONE.

### opções de fechamento

As opções que se aplicam quando um objeto é fechado O valor inicial é MQCO\_NONE. Esse atributo é ignorado durante operações de reabertura implícita, em que um valor de MQCO\_NONE é sempre usado.

### referência de conexão

Uma referência a um objeto do Gerenciador de ImqQueue fornece a conexão necessária para um gerenciador de filas (local) Para um objeto ImqQueueManager, ele é o próprio objeto. O valor inicial é zero.

**Nota:** Não confunda isso com o nome do gerenciador de filas que identifica um gerenciador de filas (possivelmente remoto) para uma fila nomeada..

### descrição

O nome descritivo (até 64 caracteres) do gerenciador de filas, fila, lista de nomes ou processo. Esse atributo é somente leitura.

### nome

O nome (até 48 caracteres) do gerenciador de filas, fila, lista de nomes ou processo. O valor inicial é uma cadeia nula. O nome de uma fila modelo muda após um **open** para o nome da fila dinâmica resultante.

**Nota:** Um gerenciador ImqQueue pode ter um nome nulo, representando o gerenciador de filas padrão. O nome muda para o gerenciador de fila real após uma abertura bem-sucedida. Uma lista ImqDistribution é dinâmica e deve ter um nome nulo.

### próximo objeto gerenciado

Este é o próximo objeto desta classe, em nenhuma ordem específica, tendo a mesma referência de conexão que este objeto. O valor inicial é zero.

### Opções de abertura

As opções que se aplicam quando um objeto é aberto O valor inicial é MQOO\_INQUIRE. Há duas maneiras de configurar valores apropriados:

1. Não configure as opções abertas e não use o método aberto. O IBM MQ ajusta automaticamente as opções de abertura e automaticamente abre, reabre e fecha objetos conforme necessário. Isso pode resultar em operações de reabertura desnecessárias, porque o IBM MQ usa o método `openFor` e isso inclui opções de abertura incrementalmente apenas
2. Configure as opções abertas antes de usar quaisquer métodos que resultem em uma chamada MQI (consulte [“Referência cruzada de C++ e MQI” na página 1848](#)). Isso assegura que as operações de reabertura desnecessárias não ocorram Configure opções de abertura explicitamente se algum dos problemas de reabertura em potencial provavelmente ocorrer (consulte [Reabrir](#)).

Se você usar o método aberto, deverá assegurar que as opções de abertura sejam apropriadas primeiro. No entanto, o uso do método aberto não é obrigatório; o IBM MQ ainda exibe o mesmo comportamento do caso 1, mas, nessa circunstância, o comportamento é eficiente.

Zero não é um valor válido; configure o valor apropriado antes de tentar abrir o objeto Isso pode ser feito usando **setOpenOptions** (*IOpenOptions*) seguido por **open** () ou **openFor** (*IRequiredOpenOption*).

### Nota:

1. MQOO\_OUTPUT é substituído por MQOO\_INQUIRE durante o método **open** para uma lista de distribuições, pois MQOO\_OUTPUT é o único **open option** válido neste momento No entanto, é uma boa prática sempre configurar MQOO\_OUTPUT explicitamente em programas de aplicativos que usam o método **open**
2. Especifique MQOO\_RESOLVE\_NAMES se você desejar usar os atributos **resolved queue manager name** e **resolved queue name** da classe

### status de abertura

Se o objeto está aberto (TRUE) ou fechado (FALSE). O valor inicial é FALSE. Esse atributo é somente leitura.

### **objeto gerenciado anterior**

O objeto anterior dessa classe, em nenhuma ordem específica, tendo a mesma referência de conexão que esse objeto O valor inicial é zero.

### **gerenciador de filas-identificador**

O identificador do gerenciador de filas. Esse atributo é somente leitura.

## **Construtores**

### **ImqObject();**

O construtor padrão..

### **ImqObject(const ImqObject & object );**

O construtor de cópia.. O status de abertura será FALSE

## **Métodos de classe (público).**

### **comportamento MQLONG estático ();**

Retorna o comportamento..

### **void setBehavior(const MQLONG behavior = 0);**

Configura o comportamento.

## **Métodos de objeto (público)**

### **operador void = (const ImqObject & object );**

Executa um fechamento se necessário e copia os dados de instância do *objeto*. O status de abertura será FALSE

### **ImqBoolean alterationDate( ImqString & date );**

Fornecer uma cópia da data de alteração Ele retornará TRUE se for bem-sucedido

### **ImqString alterationDate();**

Retorna a data de alteração sem qualquer indicação de possíveis erros

### **ImqBoolean alterationTime( ImqString & time );**

Fornecer uma cópia do tempo de alteração Ele retornará TRUE se for bem-sucedido

### **ImqString alterationTime();**

Retorna o tempo de alteração sem qualquer indicação de possíveis erros

### **ImqString alternateUserId () const;**

Retorna uma cópia do ID do usuário alternativo

### **ImqBoolean setAlternateUserId (const char \* id );**

Configura o ID do usuário alternativo O ID do usuário alternativo pode ser configurado apenas enquanto o status de abertura for FALSE Esse método retornará TRUE se for bem-sucedido

### **ImqBinary alternateSecurityId () const;**

Retorna uma cópia do ID de segurança alternativo.

### **ImqBoolean setAlternateSecurityId(const ImqBinary & );**

Configura o ID de segurança alternativo O ID de segurança alternativo pode ser configurado apenas enquanto o status de abertura for FALSE O comprimento de dados do *token* deve ser zero ou MQ\_SECURITY\_ID\_LENGTH. Ele retornará TRUE se for bem-sucedido

### **ImqBoolean setAlternateSecurityId(const MQBYTE\* token = 0);**

Configura o ID de segurança alternativo *token* pode ser zero, que é o mesmo que especificar MQSID\_NONE. Se o *token* for diferente de zero, ele deverá endereçar bytes de MQ\_SECURITY\_ID\_LENGTH de dados binários. Ao usar valores predefinidos, como MQSID\_NONE, pode ser necessário fazer uma conversão para assegurar correspondência de assinatura; por exemplo, (MQBYTE \*) MQSID\_NONE.

O ID de segurança alternativo pode ser configurado apenas enquanto o status aberto for TRUE.. Ele retornará TRUE se for bem-sucedido

**ImqBoolean setAlternateSecurityId(const unsigned char \* id = 0);**

Configura o ID de segurança alternativo

**ImqBoolean close ();**

Configura o status de abertura para FALSE Ele retornará TRUE se for bem-sucedido

**MQLONG closeOptions () const;**

Retorna as opções de fechamento

**void setCloseOpções (const MQLONG options );**

Configura as opções de fechamento..

**ImqQueueGerenciador \* connectionReference () const;**

Retorna a referência de conexão..

**void setConnectionReferência ( ImqQueueManager & manager );**

Configura a referência de conexão

**void setConnectionReferência ( ImqQueueManager \* manager = 0);**

Configura a referência de conexão

**descrição do ImqBoolean virtual ( ImqString & descrição ) = 0;**

Fornece uma cópia de uma descrição. Ele retornará TRUE se for bem-sucedido

**ImqString description ();**

Retorna uma cópia da descrição sem qualquer indicação de possíveis erros..

**virtual ImqBoolean name ( ImqString & name );**

Fornece uma cópia do nome. Ele retornará TRUE se for bem-sucedido

**ImqString name ();**

Retorna uma cópia do nome sem qualquer indicação de possíveis erros

**ImqBoolean setName (const char \* name = 0);**

Configura o nome.. O nome só pode ser configurado enquanto o status de abertura for FALSE e, para um Gerenciador de ImqQueue, enquanto o status da conexão for FALSE Ele retornará TRUE se for bem-sucedido

**ImqObject \* nextManagedObject () const;**

Retorna o próximo objeto gerenciado..

**ImqBoolean open ();**

Altera o status de abertura para TRUE abrindo o objeto conforme necessário, usando entre outros atributos as opções de abertura e o nome. Esse método usa as informações de referência de conexão e o método de conexão do Gerenciador ImqQueue, se necessário, para assegurar que o status da conexão do Gerenciador ImqQueue seja TRUE Ele retorna o status de abertura

**ImqBoolean openFor (const MQLONG required-options = 0);**

Tenta assegurar que o objeto seja aberto com opções abertas ou com opções abertas que garantem o comportamento implícito pelo valor de parâmetro *required-options* .

Se *required-options* for zero, a entrada será necessária e qualquer opção de entrada será suficiente. Portanto, se as opções de abertura já contiverem um de:

- MQOO\_INPUT\_AS\_Q\_DEF
- MQOO\_INPUT\_SHARED
- MQOO\_INPUT\_EXCLUSIVE

as opções abertas já são satisfatórias e não são alteradas; se as opções abertas ainda não contiverem nenhuma dessas opções, MQOO\_INPUT\_AS\_Q\_DEF será configurado nas opções abertas.

Se *required-options* for diferente de zero, as opções necessárias serão incluídas nas opções abertas; se *required-options* for alguma dessas opções, as outras serão reconfiguradas.

Se qualquer uma das opções de abertura for alterada e o objeto já estiver aberto, o objeto será fechado temporariamente e reaberto para ajustar as opções de abertura.

Ele retornará TRUE se for bem-sucedido Sucesso indica que o objeto está aberto com opções apropriadas.

**MQLONG openOptions () const;**

Retorna as opções abertas.

**ImqBoolean setOpenOpções (const MQLONG opções );**

Configura as opções de aberturas As opções de abertura podem ser configuradas apenas enquanto o status de abertura for FALSE Ele retornará TRUE se for bem-sucedido

**ImqBoolean openStatus () const;**

Retorna o status de abertura

**ImqObject \* previousManagedObject () const;**

Retorna o objeto gerenciado anterior..

**Identificador ImqBoolean queueManager( ImqString & id );**

Fornece uma cópia do identificador do gerenciador de filas. Ele retornará TRUE se for bem-sucedido

**ImqString queueManagerIdentificador ();**

Retorna o identificador do gerenciador de filas sem qualquer indicação de possíveis erros...

**Métodos de objeto (protegidos)****virtual ImqBoolean closeTemporarily ();**

Fecha um objeto com segurança antes de reabrir. Ele retornará TRUE se for bem-sucedido Este método assume que o status de abertura é TRUE.

**MQHCONN connectionHandle () const;**

Retorna o MQHCONN associado com a referência de conexão Esse valor será zero se não houver nenhuma referência de conexão ou se o Gerenciador não estiver conectado

**ImqBoolean consulta (const MQLONG int-attr, MQLONG & value );**

Retorna um valor inteiro, cujo índice é um valor MQIA\_\*. Em caso de erro, o valor é configurado como MQIAV\_UNDEFINED.

**Consulta ImqBoolean (const MQLONG char-attr, char \* & buffer, const size\_t length );**

Retorna uma sequência de caracteres, cujo índice é um valor de MQCA\_\*.

**Nota:** Ambos os métodos retornam apenas um único valor de atributo Se uma *captura instantânea* for necessária de mais de um valor, em que os valores são consistentes entre si por um instante, o IBM MQ C++ não fornecerá esse recurso e você deverá usar a chamada MQINQ com os parâmetros apropriados.

**virtual void openInformationDisperse ();**

Dispersa informações da seção de variáveis da estrutura de dados MQOD imediatamente após uma chamada MQOPEN.

**virtual ImqBoolean openInformationPrepare ();**

Prepara informações para a seção de variável da estrutura de dados MQOD imediatamente antes de uma chamada MQOPEN e retorna TRUE se bem-sucedida.

**ImqBoolean configurado (const MQLONG int-attr, const MQLONG value );**

Configura um atributo de número inteiro IBM MQ .

**ImqBoolean configurado (const MQLONG char-attr, const char \* buffer, const size\_t required-length );**

Configura um atributo de caractere IBM MQ

**void setNextManagedObject (const ImqObject \* object = 0);**

Configura o próximo objeto gerenciado

Atenção: Use esta função apenas se você tiver certeza de que ela não quebrará a lista de objetos gerenciados..

**void setPreviousManagedObject (const ImqObject \* object = 0);**

Configura o objeto gerenciado anterior

Atenção: Use esta função apenas se você tiver certeza de que ela não quebrará a lista de objetos gerenciados..

## Dados do objeto (protegidos).

### **MQHOBJ** *ohobj*

O identificador de objeto IBM MQ (válido apenas quando o status aberto for TRUE).

### **MQOD** *omqod*

A estrutura de dados do MQOD integrado A quantia de armazenamento alocada para essa estrutura de dados é necessária para um MQOD Versão 2. Inspeção o número da versão (*omqod.Version*) e acesse os outros campos conforme a seguir:

#### **MQOD\_VERSION\_1**

Todos os outros campos em *omqod* podem ser acessados

#### **MQOD\_VERSION\_2**

Todos os outros campos em *omqod* podem ser acessados

#### **MQOD\_VERSION\_3**

*omqod.pmqod* é um ponteiro para um MQOD dinamicamente alocado, maior. Nenhum outro campo em *omqod* pode ser acessado Todos os campos endereçados pelo *omqod.pmqod* podem ser acessados

**Nota:** *omqod.pmqod.Version* pode ser menor que *omqod.Version*, indicando que o IBM MQ MQI client tem mais funcionalidade do que o servidor IBM MQ

## Códigos de Razão

- MQRC\_ATTRIBUTE\_LOCKED
- MQRC\_INCONSISTENT\_OBJECT\_STATE
- MQRC\_NO\_CONNECTION\_REFERENCE
- MQRC\_STORAGE\_NOT\_AVAILABLE
- MQRC\_REOPEN\_SAVED\_CONTEXT\_ERR
- (códigos de razão de MQCLOSE)
- (códigos de razão de MQCONN)
- (códigos de razão de MQINQ)
- (códigos de razão de MQOPEN)
- (códigos de razão de MQSET)

## Classe C++ ImqProcess

Essa classe contém um processo de aplicativo (um objeto IBM MQ do tipo MQOT\_PROCESS) que pode ser acionado por um monitor acionador.



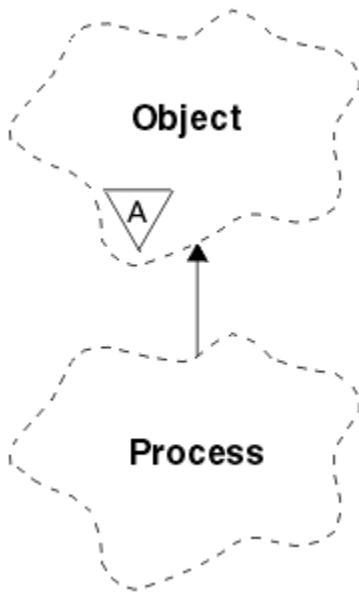


Figura 30. Classe *ImqProcess*

- [“Atributos do Objeto”](#) na página 1917
- [“Construtores”](#) na página 1917
- [“Métodos de objeto \(público\)”](#) na página 1917

## Atributos do Objeto

### ID do aplicativo

A identidade do processo de aplicativo.. Esse atributo é somente leitura.

### tipo de aplicativo

O tipo do processo de aplicativo.. Esse atributo é somente leitura.

### Dados do ambiente

As informações de ambiente do processo. Esse atributo é somente leitura.

### dados do usuário

Dados do usuário para o processo.. Esse atributo é somente leitura.

## Construtores

### **ImqProcess();**

O construtor padrão..

### **ImqProcess( const ImqProcess & process );**

O construtor de cópia.. O **ImqObject status aberto** é FALSE.

### **ImqProcess( const char \* name );**

Configura o nome do **ImqObject**

## Métodos de objeto (público)

### **void operator = ( const ImqProcess & process );**

Executa um fechamento se necessário e, em seguida, copia dados de instância do *processo*. O **ImqObject status aberto** será FALSE.

### **ImqBoolean applicationId ( ImqString e id );**

Fornecer uma cópia do **ID do aplicativo** Ele retornará TRUE se for bem-sucedido

### **ImqString applicationId ( );**

Retorna o **ID do aplicativo** sem qualquer indicação de possíveis erros

**ImqBoolean applicationType ( MQLONG & type )**

Fornecer uma cópia do **tipo de aplicativo**.. Ele retornará TRUE se for bem-sucedido

**MQLONG applicationType ();**

Retorna o **tipo de aplicativo** sem qualquer indicação de possíveis erros

**ImqBoolean environmentData ( ImqString & dados );**

Fornecer uma cópia dos **dados do ambiente** Ele retornará TRUE se for bem-sucedido

**ImqString environmentData ();**

Retorna os **dados do ambiente** sem qualquer indicação de possíveis erros

**ImqBoolean userData ( ImqString & dados )**

Fornecer uma cópia dos **dados do usuário**.. Ele retornará TRUE se for bem-sucedido

**ImqString userData ();**

Retorna os **dados do usuário** sem qualquer indicação de possíveis erros

## Classe C++ ImqPutMessageOptions

Esta classe contém a estrutura de dados MQPMO.

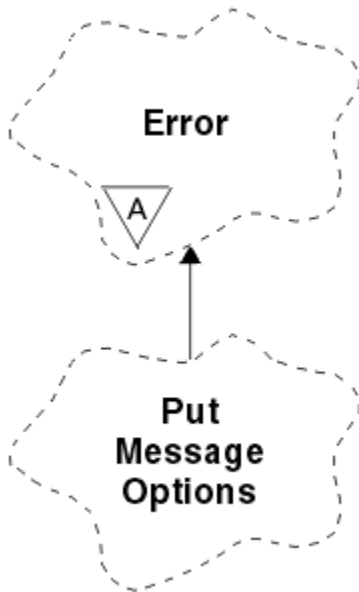


Figura 31. Classe ImqPutMessageOptions

- [“Atributos do Objeto”](#) na página 1918
- [“Construtores”](#) na página 1919
- [“Métodos de objeto \(público\)”](#) na página 1919
- [“Dados do objeto \(protegidos\).”](#) na página 1920
- [“Códigos de Razão”](#) na página 1920

### Atributos do Objeto

**referência de contexto**

Um ImqQueue que fornece um contexto para mensagens.. Inicialmente não há referência.

**opções**

As opções de mensagem de colocação O valor inicial é MQPMO\_NONE Os seguintes valores adicionais são possíveis:

- MQPMO\_SYNCPOINT
- MQPMO\_NO\_SYNCPOINT

- MQPMO\_NEW\_MSG\_ID
- MQPMO\_NEW\_CORREL\_ID
- MQPMO\_LOGICAL\_ORDER
- MQPMO\_NO\_CONTEXT
- MQPMO\_DEFAULT\_CONTEXT
- MQPMO\_PASS\_IDENTITY\_CONTEXT
- MQPMO\_PASS\_ALL\_CONTEXT
- MQPMO\_SET\_IDENTITY\_CONTEXT
- MQPMO\_SET\_ALL\_CONTEXT
- MQPMO\_ALTERNATE\_USER\_AUTHORITY
- MQPMO\_FAIL\_IF QUIESCING

### campos de registro

As sinalizações que controlam a inclusão de registros de mensagens put quando uma mensagem é colocada. O valor inicial é MQPMRF\_NONE. Os seguintes valores adicionais são possíveis:

- MQPMRF\_MSG\_ID
- MQPMRF\_CORREL\_ID
- MQPMRF\_GROUP\_ID
- MQPMRF\_FEEDBACK
- MQPMRF\_ACCOUNTING\_TOKEN

ImqMessageOs atributos do Rastreador são obtidos do objeto para qualquer campo especificado. Os atributos do Rastreador ImqMessagesão obtidos do objeto ImqMessage para qualquer campo não especificado.

### Nome do Gerenciador de Filas Resolvido

O nome de um gerenciador de filas de destino determinado durante uma colocação O valor inicial é nulo. Esse atributo é somente leitura.

### Nome da fila resolvida

Nome de uma fila de destino determinado durante um put. O valor inicial é nulo. Esse atributo é somente leitura.

### participação de ponto de sincronização

TRUE quando mensagens são colocadas sob controle de ponto de sincronização.

## Construtores

### ImqPutMessageOptions( );

O construtor padrão..

### ImqPutMessageOptions(const ImqPutMessageOptions & pmo )

O construtor de cópia..

## Métodos de objeto (público)

### void operator = (const ImqPutMessageOptions & pmo );

Copia os dados de instância do *pmo*, substituindo os dados de instância existentes

### ImqQueue \* contextReference () const;

Retorna a referência de contexto..

### void setContextReference (const ImqQueue & queue );

Configura a referência de contexto

### void setContextReferência (const ImqQueue \* queue = 0);

Configura a referência de contexto

**Opções MQLONG () const;**

Retorna as opções..

**void setOptions (const MQLONG options );**

Configura as opções, incluindo o valor de participação do ponto de sincronização

**MQLONG recordFields () const;**

Retorna os campos de registro..

**void setRecordCampos (const MQLONG campos );**

Configura os campos de registro

**ImqString resolvedQueueManagerName (). const;**

Retorna uma cópia do nome do gerenciador de filas resolvido

**ImqString resolvedQueueNome () const;**

Retorna uma cópia do nome da fila resolvida

**ImqBoolean syncPointParticipação () const;**

Retorna o valor de participação do ponto de sincronização, que será TRUE se as opções incluírem MQPMO\_SYNCPOINT..

**void setSyncPointParticipation (const ImqBoolean sync );**

Configura o valor de participação do ponto de sincronização. Se *sync* for TRUE, as opções serão alteradas para incluir MQPMO\_SYNCPOINT e excluir MQPMO\_NO\_SYNCPOINT. Se *sync* for FALSE, as opções serão alteradas para incluir MQPMO\_NO\_SYNCPOINT e excluir MQPMO\_SYNCPOINT.

**Dados do objeto (protegidos).****MQPMO omqpmo**

A estrutura de dados do MQPMO

**Códigos de Razão**

- MQRC\_STORAGE\_NOT\_AVAILABLE

**Classe C++ ImqQueue**

Essa classe contém uma fila de mensagens (um objeto IBM MQ do tipo MQOT\_Q).

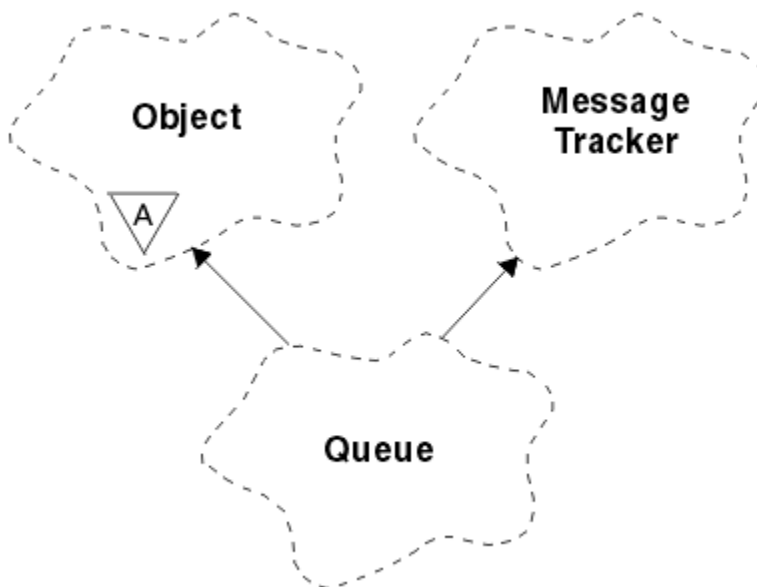


Figura 32. Classe ImqQueue

Essa classe está relacionada às chamadas MQI listadas em [Tabela 862 na página 1857](#).

- [“Atributos do Objeto” na página 1921](#)
- [“Construtores” na página 1924](#)
- [“Métodos de objeto \(público\)” na página 1924](#)
- [“Métodos de objeto \(protegidos\)” na página 1930](#)
- [“Códigos de Razão” na página 1930](#)

## **Atributos do Objeto**

### **Nome do reenfileiramento de restauração**

Nome de reenfileiramento de restauração em excesso Esse atributo é somente leitura.

### **limite de restauração**

Limite de restauração. Esse atributo é somente leitura.

### **Nome da Fila Base**

Nome da fila para a qual o alias é resolvido. Esse atributo é somente leitura.

### **nome do cluster**

Nome do cluster. Esse atributo é somente leitura.

### **Nome da lista de nomes do cluster**

Nome da lista de nomes do cluster Esse atributo é somente leitura.

### **Classificação da Carga de Trabalho de Cluster**

Classificação da carga de trabalho de cluster. Esse atributo é somente leitura.

### **Prioridade da Carga de Trabalho de Cluster**

Prioridade da carga de trabalho de cluster. Esse atributo é somente leitura.

### **Fila de uso de carga de trabalho do cluster**

A carga de trabalho do cluster usa o valor da fila. Esse atributo é somente leitura.

### **data de criação**

Dados de criação da fila Esse atributo é somente leitura.

### **horário de criação**

Horário de criação da fila Esse atributo é somente leitura.

### **Profundidade atual**

O número de mensagens na fila Esse atributo é somente leitura.

### **Ligação Padrão**

Ligação padrão.. Esse atributo é somente leitura.

### **Opções de abertura de entrada padrão**

Opção open-for-input padrão. Esse atributo é somente leitura.

### **Persistência padrão**

Persistência de mensagem padrão.. Esse atributo é somente leitura.

### **Prioridade padrão**

Prioridade da mensagem padrão. Esse atributo é somente leitura.

### **tipo de definição**

O tipo de definição de fila. Esse atributo é somente leitura.

### **evento de alta profundidade**

Atributo de controle para eventos de alta profundidade da fila Esse atributo é somente leitura.

### **Limite superior de profundidade**

Limite alto para a profundidade da fila Esse atributo é somente leitura.

### **evento de baixa profundidade**

Atributo de controle para eventos de baixa profundidade da fila Esse atributo é somente leitura.

### **Limite inferior de profundidade**

Limite baixo para a profundidade da fila Esse atributo é somente leitura.

### **evento máximo de profundidade**

Atributo de controle para eventos máximos de profundidade da fila Esse atributo é somente leitura.

**referência da lista de distribuições**

Referência opcional para uma Lista ImqDistribution que pode ser usada para distribuir mensagens para mais de uma fila, incluindo esta. O valor inicial é nulo.

**Nota:** Quando um objeto ImqQueue é aberto, qualquer objeto da Lista ImqDistribution aberto que ele referenciar é automaticamente fechado

**listas de distribuição**

A capacidade de uma fila de transmissão suportar listas de distribuição. Esse atributo é somente leitura.

**Nome da fila dinâmica**

Nome da fila dinâmica.. O valor inicial é AMQ.\* para todas as plataformas do AIX, Linux, and Windows

**Forçar obtenção de recuperação**

Se a contagem de restaurações deve ser endurecida. Esse atributo é somente leitura.

**Tipo de índice**

Tipo de índice.. Esse atributo é somente leitura.

**inibe obter**

Se operações get são permitidas. O valor inicial é dependente da definição de fila. Este atributo é válido apenas para um alias ou fila local..

**inibir colocação**

Se operações put são permitidas. O valor inicial é dependente da definição de fila.

**Nome da Fila de Inicialização**

Nome da fila de iniciação Esse atributo é somente leitura.

**Profundidade máxima**

Número máximo de mensagens permitidas na fila.. Esse atributo é somente leitura.

**comprimento máximo da mensagem**

Comprimento máximo para qualquer mensagem nessa fila, que pode ser menor do que o máximo para qualquer fila gerenciada pelo gerenciador de filas associado Esse atributo é somente leitura.

**Seqüência de entrega de mensagens**

Indica se a prioridade da mensagem é relevante Esse atributo é somente leitura.

**próxima fila distribuída**

O próximo objeto dessa classe, em nenhuma ordem específica, tendo a mesma **referência da lista de distribuição** que esse objeto O valor inicial é zero.

Se um objeto em uma cadeia for excluído, o objeto anterior e o próximo objeto serão atualizados para que seus vínculos de fila distribuídos não apontem mais para o objeto excluído

**classe de mensagem não persistente**

Nível de confiabilidade para mensagens não persistentes enviadas para esta fila Esse atributo é somente leitura.

**Contagem de entradas abertas**

Número de objetos ImqQueue abertos para entrada. Esse atributo é somente leitura.

**Contagem de saídas abertas**

Número de objetos ImqQueue abertos para saída. Esse atributo é somente leitura.

**fila distribuída anterior**

O objeto anterior dessa classe, em nenhuma ordem específica, tendo a mesma **referência da lista de distribuição** que esse objeto O valor inicial é zero.

Se um objeto em uma cadeia for excluído, o objeto anterior e o próximo objeto serão atualizados para que seus vínculos de fila distribuídos não apontem mais para o objeto excluído

**nome do processo**

Nome da definição de processo.. Esse atributo é somente leitura.

**Enfileiramento de Contas**

Nível de informações de contabilidade para filas. Esse atributo é somente leitura.

**nome do gerenciador de filas**

Nome do gerenciador de filas (possivelmente remoto) no qual a fila reside. Não confunda o gerenciador de filas nomeado aqui com a ImqObject **referência de conexão**, que faz referência ao gerenciador de filas (local) que fornece uma conexão... O valor inicial é nulo.

**Monitoramento de Fila**

Nível de coleta de Dados de Monitoramento para a fila. Esse atributo é somente leitura.

**estatísticas de fila**

Nível de dados de estatísticas para a fila. Esse atributo é somente leitura.

**tipo de fila**

Tipo de fila. Esse atributo é somente leitura.

**Nome do Gerenciador de Filas Remotas**

Nome do gerenciador de filas remotas. Esse atributo é somente leitura.

**Nome da fila remota**

Nome da fila remota, conforme conhecido no gerenciador de filas remotas. Esse atributo é somente leitura.

**Nome do Gerenciador de Filas Resolvido**

Nome do gerenciador de filas resolvido. Esse atributo é somente leitura.

**Nome da fila resolvida**

Nome da fila resolvida. Esse atributo é somente leitura.

**intervalo de retenção**

Intervalo de retenção da fila. Esse atributo é somente leitura.

**escopo**

Escopo da definição de fila.. Esse atributo é somente leitura.

**intervalo de serviço**

Intervalo de serviço. Esse atributo é somente leitura.

**evento de intervalo de serviço**

Atributo de controle para eventos de intervalos de serviço. Esse atributo é somente leitura.

**Capacidade de compartilhamento**

Se a fila pode ser compartilhada. Esse atributo é somente leitura.

**classe de armazenamento**

Classe de armazenamento.. Esse atributo é somente leitura.

**Nome da fila de transmissão**

Nome da fila de transmissão. Esse atributo é somente leitura.

**Controle de ativação**

Acionador de controle. O valor inicial depende da definição de fila. Esse atributo é válido apenas para uma fila local.

**dados do acionador**

Dados do acionador. O valor inicial depende da definição de fila. Esse atributo é válido apenas para uma fila local.

**Profundidade de ativação**

Profundidade do acionador. O valor inicial depende da definição de fila. Esse atributo é válido apenas para uma fila local.

**Prioridade de mensagem de ativação**

Prioridade da mensagem limite para acionadores. O valor inicial depende da definição de fila. Esse atributo é válido apenas para uma fila local.

**tipo de acionador**

Tipo de acionador. O valor inicial depende da definição de fila. Esse atributo é válido apenas para uma fila local.

**uso**

Uso. Esse atributo é somente leitura.

## Construtores

### **ImqQueue();**

O construtor padrão..

### **ImqQueue( const ImqQueue & );**

O construtor de cópia.. O ImqObject **status aberto** será FALSE.

### **ImqQueue( const char \* name );**

Configura o nome do ImqObject

## Métodos de objeto (público)

### **void operator = ( const ImqQueue & queue );**

Executa um fechamento se necessário e, em seguida, copia dados de instância da *fila*. O ImqObject **status aberto** será FALSE.

### **ImqBoolean backoutRequeueName ( ImqString e name );**

Fornecer uma cópia do **nome de refila de restauração**. Ele retornará TRUE se for bem-sucedido

### **ImqString backoutRequeueName ();**

Retorna o **backout requeue name** sem qualquer indicação de possíveis erros.

### **ImqBoolean backoutThreshold ( MQLONG & limite );**

Fornecer uma cópia do **limite de restauração**. Ele retornará TRUE se for bem-sucedido

### **MQLONG backoutThreshold ();**

Retorna o valor do **limite de restauração** sem qualquer indicação de possíveis erros

### **ImqBoolean baseQueueNome ( ImqString & nome );**

Fornecer uma cópia do **nome da fila base**. Ele retornará TRUE se for bem-sucedido

### **ImqString baseQueueName ();**

Retorna o **nome da fila base** sem qualquer indicação de possíveis erros..

### **ImqBoolean clusterName( ImqString & name );**

Fornecer uma cópia do **nome do cluster** Ele retornará TRUE se for bem-sucedido

### **ImqString clusterName();**

Retorna o **nome do cluster** sem qualquer indicação de possíveis erros

### **ImqBoolean clusterNamelistNome ( ImqString & nome );**

Fornecer uma cópia do **nome da lista de nomes do cluster** Ele retornará TRUE se for bem-sucedido

### **ImqString clusterNamelistName ();**

Retorna o **nome da lista de nomes do cluster** sem qualquer indicação de erros

### **ImqBoolean clusterWorkLoadPriority (MQLONG & priority);**

Fornecer uma cópia do valor de prioridade de carga de trabalho do cluster Ele retornará TRUE se for bem-sucedido

### **MQLONG clusterWorkLoadPriority ();**

Retorna o valor de prioridade da carga de trabalho do cluster sem qualquer indicação de possíveis erros..

### **ImqBoolean clusterWorkLoadRank (MQLONG e classificação);**

Fornecer uma cópia do valor de classificação de carga de trabalho do cluster Ele retornará TRUE se for bem-sucedido

### **MQLONG clusterWorkLoadRank ();**

Retorna o valor de classificação da carga de trabalho do cluster sem qualquer indicação de possíveis erros

### **ImqBoolean clusterWorkLoadUseQ (MQLONG & useq);**

Fornecer uma cópia do valor da fila de uso da carga de trabalho do cluster Ele retornará TRUE se for bem-sucedido

### **MQLONG clusterWorkLoadUseQ ();**

Retorna o valor da fila de uso da carga de trabalho do cluster sem nenhuma indicação de possíveis erros..



**ImqBoolean creationDate ( ImqString & date );**  
Fornece uma cópia da **data de criação** Ele retornará TRUE se for bem-sucedido

**ImqString creationDate ( );**  
Retorna a **data de criação** sem qualquer indicação de possíveis erros

**ImqBoolean creationTime ( ImqString & time );**  
Fornece uma cópia do **horário de criação** Ele retornará TRUE se for bem-sucedido

**ImqString creationTime ( );**  
Retorna o **horário de criação** sem qualquer indicação de possíveis erros

**ImqBoolean currentDepth ( MQLONG & profundidade )**  
Fornece uma cópia da **profundidade atual** Ele retornará TRUE se for bem-sucedido

**MQLONG currentDepth ();**  
Retorna a **profundidade atual** sem qualquer indicação de possíveis erros

**ImqBoolean defaultInputOpenOption ( MQLONG & opção ).**  
Fornece uma cópia da **opção de abertura de entrada padrão** Ele retornará TRUE se for bem-sucedido

**MQLONG defaultInputOpenOption ();**  
Retorna a **opção de abertura de entrada padrão** sem qualquer indicação de possíveis erros

**ImqBoolean defaultPersistence ( MQLONG & persistence );**  
Fornece uma cópia da **persistência padrão**.. Ele retornará TRUE se for bem-sucedido

**MQLONG defaultPersistence ();**  
Retorna a **persistência padrão** sem qualquer indicação de possíveis erros

**ImqBoolean defaultPriority ( MQLONG & prioridade );**  
Fornece uma cópia da **prioridade padrão**.. Ele retornará TRUE se for bem-sucedido

**MQLONG defaultPriority ();**  
Retorna a **prioridade padrão** sem qualquer indicação de possíveis erros

**ImqBoolean defaultBind ( MQLONG & bind );**  
Fornece uma cópia da **ligação padrão**.. Ele retornará TRUE se for bem-sucedido

**MQLONG defaultBind ();**  
Retorna a **ligação padrão** sem qualquer indicação de possíveis erros

**ImqBoolean definitionType ( MQLONG & tipo );**  
Fornece uma cópia do **tipo de definição** Ele retornará TRUE se for bem-sucedido

**MQLONG definitionType ();**  
Retorna o **tipo de definição** sem qualquer indicação de possíveis erros

**ImqBoolean depthHighEvento ( MQLONG & evento );**  
Fornece uma cópia do estado de ativação do **evento de profundidade alta** Ele retornará TRUE se for bem-sucedido

**MQLONG depthHighEvent ();**  
Retorna o estado de ativação do **evento de alta profundidade** sem qualquer indicação de possíveis erros

**ImqBoolean depthHighLimit ( MQLONG & limit );**  
Fornece uma cópia do **limite alto de profundidade**.. Ele retornará TRUE se for bem-sucedido

**MQLONG depthHighLimit ();**  
Retorna o valor de **limite alto de profundidade** sem qualquer indicação de possíveis erros.

**ImqBoolean depthLowEvento ( MQLONG & evento );**  
Fornece uma cópia de um estado de ativação do **evento de baixa profundidade** Ele retornará TRUE se for bem-sucedido

**MQLONG depthLowEvent ();**  
Retorna o estado de ativação do **evento de baixa profundidade** sem qualquer indicação de possíveis erros

**ImqBoolean depthLowLimite ( MQLONG & limite );**

Fornecer uma cópia do **limite baixo de profundidade** Ele retornará TRUE se for bem-sucedido

**MQLONG depthLowLimit ();**

Retorna o valor de **limite baixo de profundidade** sem qualquer indicação de possíveis erros

**ImqBoolean depthMaximumEvent ( MQLONG & event );**

Fornecer uma cópia do estado de ativação do **evento máximo de profundidade**.. Ele retornará TRUE se for bem-sucedido

**MQLONG depthMaximumEvent ();**

Retorna o estado de ativação do **evento de profundidade máxima** sem qualquer indicação de possíveis erros

**ImqDistributionLista \* distributionListReference () const ;**

Retorna **referência da lista de distribuição**.

**void setDistributionListReference ( ImqDistributionList & list );**

Configura a **referência da lista de distribuição**..

**void setDistributionListReference ( ImqDistributionList \* list = 0);**

Configura a **referência da lista de distribuição**..

**ImqBoolean distributionLists ( MQLONG & support );**

Fornecer uma cópia do valor **distribution lists** . Ele retornará TRUE se for bem-sucedido

**MQLONG distributionLists ();**

Retorna o valor **distribution lists** sem qualquer indicação de possíveis erros.

**ImqBoolean setDistributionLists ( const MQLONG support );**

Configura o valor de **listas de distribuição** . Ele retornará TRUE se for bem-sucedido

**ImqString dynamicQueueName () const ;**

Retorna uma cópia do **nome da fila dinâmica**

**ImqBoolean setDynamicQueueName ( const char \* name );**

Configura o **nome da fila dinâmica** O **nome da fila dinâmica** pode ser configurado apenas enquanto o ImqObject **status de abertura** for FALSE Ele retornará TRUE se for bem-sucedido

**ImqBoolean get ( ImqMessage & msg, ImqGetMessageOptions & opções );**

Recupera uma mensagem da fila, usando as **opções** especificadas. Chama o método ImqObject **openFor** se necessário para assegurar que o ImqObject **open options** inclua um dos valores MQOO\_INPUT\_ \* ou o valor MQOO\_BROWSE, dependendo das **opções**. Se o objeto *msg* tiver um ImqCache **buffer automático**, o buffer crescerá para acomodar qualquer mensagem recuperada O método **clearMessage** é chamado no objeto *msg* antes da recuperação.

Esse método retornará TRUE se for bem-sucedido

**Nota:** O resultado da chamada de método será FALSE se o código de razão ImqObject for MQRC\_TRUNCATED\_MSG\_FAILED, mesmo que esse **código de razão** seja classificado como um aviso Se uma mensagem truncada for aceita, o ImqCache **comprimento da mensagem** reflete o comprimento truncado. Em qualquer evento, o ImqMessage **comprimento total da mensagem** indica o número de bytes que estavam disponíveis.

**ImqBoolean get ( ImqMessage & msg );**

Como para o método anterior, exceto que as opções de mensagem de obtenção padrão são usadas

**ImqBoolean get ( ImqMessage & msg, ImqGetMessageOptions & options, const size\_t buffer-size );**

Como para os dois métodos anteriores, exceto que uma substituição de *buffer-size* é indicada. Se o objeto *msg* usar um ImqCache **buffer automático**, o método **resizeBuffer** será chamado no objeto *msg* antes da recuperação da mensagem e o buffer não aumentará ainda mais para acomodar qualquer mensagem maior.

**ImqBoolean get ( ImqMessage & msg, const size\_t buffer-size );**

Como para o método anterior, exceto que as opções de mensagem de obtenção padrão são usadas

**ImqBoolean hardenGetBackout ( MQLONG & harden );**

Fornecer uma cópia do valor **harden get backout** . Ele retornará TRUE se for bem-sucedido

**MQLONG hardenGetBackout ();**

Retorna o valor de **harden get backout** sem qualquer indicação de possíveis erros.

**ImqBoolean indexType(MQLONG & type );**

Fornece uma cópia do **tipo de índice** Ele retornará TRUE se for bem-sucedido

**MQLONG indexType();**

Retorna o **tipo de índice** sem qualquer indicação de possíveis erros

**ImqBoolean inhibitGet ( MQLONG & inibe );**

Fornece uma cópia do valor **inibir obtenção** . Ele retornará TRUE se for bem-sucedido

**MQLONG inhibitGet ();**

Retorna o valor **inibir obter** sem qualquer indicação de possíveis erros

**ImqBoolean setInhibitGet ( const MQLONG inibe );**

Configura o valor **inibe get** . Ele retornará TRUE se for bem-sucedido

**ImqBoolean inhibitPut ( MQLONG & inibe );**

Fornece uma cópia do valor **inibe o put** . Ele retornará TRUE se for bem-sucedido

**MQLONG inhibitPut ();**

Retorna o valor **inibir put** sem qualquer indicação de possíveis erros

**ImqBoolean setInhibitPut ( const MQLONG inibe );**

Configura o valor **inibe put** . Ele retornará TRUE se for bem-sucedido

**ImqBoolean initiationQueueName ( ImqString & name );**

Fornece uma cópia do **nome da fila de inicialização**.. Ele retornará TRUE se for bem-sucedido

**ImqString initiationQueueName ();**

Retorna o **nome da fila de iniciação** sem qualquer indicação de possíveis erros

**ImqBoolean maximumDepth ( MQLONG & profundidade );**

Fornece uma cópia da **profundidade máxima** Ele retornará TRUE se for bem-sucedido

**MQLONG maximumDepth (),**

Retorna a **profundidade máxima** sem qualquer indicação de possíveis erros

**ImqBoolean maximumMessageLength ( MQLONG & length );**

Fornece uma cópia do **comprimento máximo da mensagem** Ele retornará TRUE se for bem-sucedido

**MQLONG maximumMessageLength ();**

Retorna o **comprimento máximo da mensagem** sem qualquer indicação de possíveis erros

**ImqBoolean messageDeliverySequence ( MQLONG & sequence );**

Fornece uma cópia do **message delivery sequence**. Ele retornará TRUE se for bem-sucedido

**MQLONG messageDeliverySequence ();**

Retorna o valor **message delivery sequence** sem qualquer indicação de possíveis erros.

**ImqQueue \* nextDistributedQueue () const ;**

Retorna a **próxima fila distribuída**

**ImqBoolean nonPersistentMessageClass (MQLONG & monq);**

Fornece uma cópia do valor da classe de mensagem não persistente Ele retornará TRUE se for bem-sucedido

**MQLONG nonPersistentMessageClass ();**

Retorna o valor da classe de mensagem não persistente sem qualquer indicação de possíveis erros..

**ImqBoolean openInputCount ( MQLONG & count );**

Fornece uma cópia da **contagem de entrada de abertura** Ele retornará TRUE se for bem-sucedido

**MQLONG openInputCount ();**

Retorna a **contagem de entrada de abertura** sem qualquer indicação de possíveis erros

**ImqBoolean openOutputCount ( MQLONG & count );**

Fornece uma cópia do **open output count**. Ele retornará TRUE se for bem-sucedido

**MQLONG openOutputCount ();**

Retorna a **contagem de saída de abertura** sem qualquer indicação de possíveis erros

**ImqQueue \* previousDistributedQueue () const ;**

Retorna a **fila distribuída anterior**

**ImqBoolean processName ( ImqString e name );**

Fornece uma cópia do **nome do processo** Ele retornará TRUE se for bem-sucedido

**ImqString processName ();**

Retorna o **nome do processo** sem qualquer indicação de possíveis erros

**ImqBoolean put ( ImqMessage & msg );**

Coloca uma mensagem na fila, usando opções de mensagem put padrão. Usa o método ImqObject **openFor** se necessário para assegurar que as ImqObject **opções abertas** incluam MQOO\_OUTPUT.

Esse método retornará TRUE se for bem-sucedido

**ImqBoolean put ( ImqMessage & msg, ImqPutMessageOptions & pmo );**

Coloca uma mensagem na fila, usando o *pmo* especificado. Usa o método ImqObject **openFor** conforme necessário para assegurar que o ImqObject **open options** inclua MQOO\_OUTPUT e (se as opções *pmo* incluírem qualquer um dos MQPMO\_PASS\_IDENTITY\_CONTEXT, MQPMO\_PASS\_ALL\_CONTEXT, MQPMO\_SET\_IDENTITY\_CONTEXT ou MQPMO\_SET\_ALL\_CONTEXT) valores MQOO\_\*\_CONTEXT correspondentes.

Esse método retornará TRUE se for bem-sucedido

**Nota:** Se o *pmo* incluir uma **referência de contexto**, o objeto de referência será aberto, se necessário, para fornecer um contexto.

**ImqBoolean queueAccounting (MQLONG & acctq)**

Fornece uma cópia do valor de contabilidade da fila. Ele retornará TRUE se for bem-sucedido

**MQLONG queueAccounting ();**

Retorna o valor contábil da fila sem qualquer indicação de possíveis erros

**ImqString queueManagerName () const ;**

Retorna o **nome do gerenciador de filas..**

**ImqBoolean setQueueManagerName ( const char \* nome );**

Configura o **nome do gerenciador de filas**. O **nome do gerenciador de filas** pode ser configurado apenas enquanto o ImqObject **status de abertura** for FALSE Esse método retornará TRUE se for bem-sucedido

**ImqBoolean queueMonitoring (MQLONG & monq);**

Fornece uma cópia do valor de monitoramento de fila. Ele retornará TRUE se for bem-sucedido

**MQLONG queueMonitoring ();**

Retorna o valor de monitoramento da fila sem qualquer indicação de erros possíveis

**ImqBoolean queueStatistics (MQLONG & statq)**

Fornece uma cópia do valor de estatísticas de fila. Ele retornará TRUE se for bem-sucedido

**MQLONG queueStatistics ();**

Retorna o valor de estatísticas de fila sem qualquer indicação de possíveis erros..

**ImqBoolean queueType ( MQLONG & tipo );**

Fornece uma cópia do valor **queue type** . Ele retornará TRUE se for bem-sucedido

**MQLONG queueType ();**

Retorna o **tipo de fila** sem qualquer indicação de possíveis erros

**ImqBoolean remoteQueueManagerName ( ImqString e name );**

Fornece uma cópia do **nome do gerenciador de filas remotas** Ele retornará TRUE se for bem-sucedido

**ImqString remoteQueueManagerName ();**

Retorna o **nome do gerenciador de filas remotas** sem qualquer indicação de possíveis erros

**ImqBoolean remoteQueueNome ( ImqString & nome );**

Fornece uma cópia do **nome da fila remota** Ele retornará TRUE se for bem-sucedido

**ImqString remoteQueueName ();**

Retorna o **nome da fila remota** sem qualquer indicação de possíveis erros

**ImqBoolean resolvedQueueManagerName( ImqString & name ).**

Fornece uma cópia do **nome do gerenciador de filas resolvido**. Ele retornará TRUE se for bem-sucedido

**Nota:** Esse método falha, a menos que MQOO\_RESOLVE\_NAMES esteja entre as opções de abertura ImqObject

**ImqString resolvedQueueManagerName( ) ;**

Retorna o **nome do gerenciador de filas resolvido**, sem qualquer indicação de possíveis erros

**ImqBoolean resolvedQueueNome ( ImqString & nome );**

Fornece uma cópia do **nome da Fila Resolvida** Ele retornará TRUE se for bem-sucedido

**Nota:** Esse método falha, a menos que MQOO\_RESOLVE\_NAMES esteja entre as opções de abertura ImqObject

**ImqString resolvedQueueName ();**

Retorna o **nome da fila resolvida**, sem qualquer indicação de possíveis erros

**ImqBoolean retentionInterval ( MQLONG & interval );**

Fornece uma cópia do **intervalo de retenção**. Ele retornará TRUE se for bem-sucedido

**MQLONG retentionInterval ();**

Retorna o **intervalo de retenção** sem qualquer indicação de possíveis erros.

**ImqBoolean scope ( MQLONG & scope );**

Fornece uma cópia do **escopo** Ele retornará TRUE se for bem-sucedido

**MQLONG scope ();**

Retorna o **escopo** sem qualquer indicação de possíveis erros

**ImqBoolean serviceInterval ( MQLONG & interval );**

Fornece uma cópia do **intervalo de serviço**. Ele retornará TRUE se for bem-sucedido

**MQLONG serviceInterval ();**

Retorna o **intervalo de serviço** sem qualquer indicação de possíveis erros..

**ImqBoolean serviceIntervalEvento ( MQLONG & evento )**

Fornece uma cópia do estado de ativação do **evento de intervalo de serviço**.. Ele retornará TRUE se for bem-sucedido

**MQLONG serviceIntervalEvent ();**

Retorna o estado de ativação do **service interval event** sem qualquer indicação de possíveis erros.

**ImqBoolean shareability ( MQLONG & shareability );**

Fornece uma cópia do valor **shareability** . Ele retornará TRUE se for bem-sucedido

**MQLONG shareability ();**

Retorna o valor **shareability** sem qualquer indicação de possíveis erros.

**ImqBoolean storageClass( ImqString & class );**

Fornece uma cópia da **classe de armazenamento** Ele retornará TRUE se for bem-sucedido

**ImqString storageClass( );**

Retorna a **classe de armazenamento** sem qualquer indicação de possíveis erros

**ImqBoolean transmissionQueueName ( ImqString & name );**

Fornece uma cópia do **nome da fila de transmissão**.. Ele retornará TRUE se for bem-sucedido

**ImqString transmissionQueueName ();**

Retorna o **nome da fila de transmissão** sem qualquer indicação de possíveis erros..

**ImqBoolean triggerControl ( MQLONG & control );**

Fornece uma cópia do valor **trigger control** . Ele retornará TRUE se for bem-sucedido

**MQLONG triggerControl ();**

Retorna o valor **trigger control** sem qualquer indicação de possíveis erros.

**ImqBoolean setTriggerControl ( const MQLONG control );**

Configura o valor de **controle do acionador** . Ele retornará TRUE se for bem-sucedido

**ImqBoolean triggerData ( ImqString & dados );**

Fornecer uma cópia dos **dados do acionador** Ele retornará TRUE se for bem-sucedido

**ImqString triggerData ( );**

Retorna uma cópia dos **dados do acionador** sem qualquer indicação de possíveis erros

**ImqBoolean setTriggerData ( const char \* data );**

Configura os **dados do acionador** Ele retornará TRUE se for bem-sucedido

**ImqBoolean triggerDepth ( MQLONG & profundidade );**

Fornecer uma cópia da **profundidade do acionador** Ele retornará TRUE se for bem-sucedido

**MQLONG triggerDepth ();**

Retorna a **profundidade do acionador** sem qualquer indicação de possíveis erros

**ImqBoolean setTriggerDepth ( const MQLONG depth );**

Configura a **profundidade do acionador** Ele retornará TRUE se for bem-sucedido

**ImqBoolean triggerMessagePriority ( MQLONG & priority );**

Fornecer uma cópia da **prioridade da mensagem do acionador** Ele retornará TRUE se for bem-sucedido

**MQLONG triggerMessagePriority ();**

Retorna a **prioridade da mensagem do acionador** sem qualquer indicação de possíveis erros

**ImqBoolean setTriggerMessagePriority ( const MQLONG priority );**

Configura a **prioridade da mensagem do acionador** Ele retornará TRUE se for bem-sucedido

**ImqBoolean triggerType ( MQLONG & type );**

Fornecer uma cópia do **tipo de acionador**.. Ele retornará TRUE se for bem-sucedido

**MQLONG triggerType ();**

Retorna o **tipo de acionador** sem qualquer indicação de possíveis erros

**ImqBoolean setTriggerTipo ( const MQLONG tipo );**

Configura o **tipo de acionador** Ele retornará TRUE se for bem-sucedido

**ImqBoolean uso ( MQLONG & uso );**

Fornecer uma cópia do valor **usage** . Ele retornará TRUE se for bem-sucedido

**MQLONG uso ();**

Retorna o valor **usage** sem qualquer indicação de possíveis erros.

**Métodos de objeto (protegidos)****void setNextDistributedQueue ( ImqQueue \* queue = 0);**

Configura a **próxima fila distribuída**

**Atenção:** Use esta função apenas se tiver certeza de que ela não quebrará a lista de filas distribuídas.

**void setPreviousDistributedQueue ( ImqQueue \* queue = 0);**

Configura a **fila distribuída anterior**

**Atenção:** Use esta função apenas se tiver certeza de que ela não quebrará a lista de filas distribuídas.

**Códigos de Razão**

- MQRC\_ATTRIBUTE\_LOCKED
- MQRC\_CONTEXT\_OBJECT\_NOT\_VALID
- MQRC\_CONTEXT\_OPEN\_ERROR
- MQRC\_CURSOR\_NOT\_VALID
- MQRC\_NO\_BUFFER
- MQRC\_REOPEN\_EXCL\_INPUT\_ERROR
- MQRC\_REOPEN\_INQUIRE\_ERROR
- MQRC\_REOPEN\_TEMPORARY\_Q\_ERROR

- (códigos de razão de MQGET)
- (códigos de razão de MQPUT)

## Classe C++ do Gerenciador ImqQueue

Essa classe contém um gerenciador de filas (um objeto IBM MQ do tipo MQOT\_Q\_MGR).

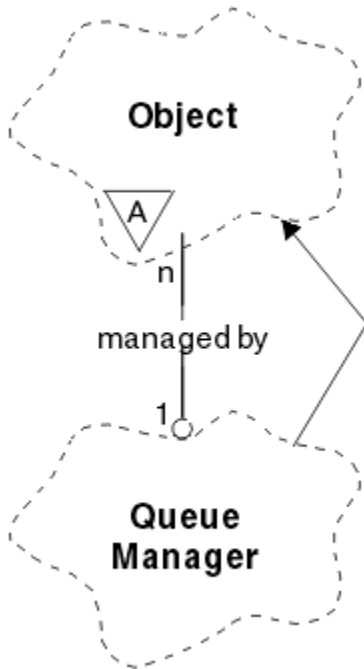


Figura 33. Classe do Gerenciador ImqQueue

Essa classe está relacionada às chamadas MQI listadas em [“Referência cruzada do Gerenciador ImqQueue”](#) na página 1859. Nem todos os métodos listados são aplicáveis a todas as plataformas; consulte [ALTER QMGR](#) para obter mais detalhes

- [“Atributos de classe.”](#) na página 1931
- [“Atributos do Objeto”](#) na página 1932
- [“Construtores”](#) na página 1937
- [“Destruidores”](#) na página 1937
- [“Métodos de classe \(público\).”](#) na página 1937
- [“Métodos de objeto \(público\)”](#) na página 1938
- [“Métodos de objeto \(protegidos\)”](#) na página 1947
- [“Dados do objeto \(protegidos\).”](#) na página 1947
- [“Códigos de Razão”](#) na página 1947

### Atributos de classe.

#### comportamento

Controla o comportamento de conexão implícita e desconexão.

#### **IMQ\_EXPL\_DISC\_BACKOUT (0L)**

Uma chamada explícita para o método de desconexão implica restauração. Esse atributo é mutuamente exclusivo com IMQ\_EXPL\_DISC\_COMMIT..

#### **IMQ\_EXPL\_DISC\_COMMIT (1L)**

Uma chamada explícita para o método de desconexão implica em confirmação (o padrão) Esse atributo é mutuamente exclusivo com IMQ\_EXPL\_DISC\_BACKOUT..

**IMQ\_IMPL\_CONN (2L)**

Conexão implícita é permitida (o padrão).

**IMQ\_IMPL\_DISC\_BACKOUT (0L)**

Uma chamada implícita ao método de desconexão, que pode ocorrer durante a destruição do objeto, implica em restauração. Esse atributo é mutuamente exclusivo com o IMQ\_IMPL\_DISC\_COMMIT

**IMQ\_IMPL\_DISC\_COMMIT (4L)**

Uma chamada implícita para o método de desconexão, que pode ocorrer durante a destruição do objeto, implica commit (o padrão). Esse atributo é mutuamente exclusivo com IMQ\_IMPL\_DISC\_BACKOUT..

No IBM MQ V7.0 e acima, os aplicativos C++ que usam uma conexão implícita precisam especificar IMQ\_IMPL\_CONN juntamente com quaisquer outras opções fornecidas no método `setBehavior()` em um objeto de classe `ImqQueueManager`. Se seu aplicativo não usar o método `setBehavior()` para configurar explicitamente as opções de comportamento, por exemplo,

```
ImqQueueManager_object.setBehavior(IMQ_IMPL_DISC_COMMIT)
```

essa mudança não afeta você, pois MQ\_IMPL\_CONN está ativado por padrão.

Se seu aplicativo configurar explicitamente as opções de comportamento, por exemplo,

```
ImqQueueManager_object.setBehavior(IMQ_IMPL_DISC_COMMIT)
```

é necessário incluir IMQ\_IMPL\_CONN no método `setBehavior()` da seguinte forma, para permitir que seu aplicativo conclua uma conexão implícita:

```
ImqQueueManager_object.setBehavior(IMQ_IMPL_CONN | IMQ_IMPL_DISC_COMMIT)
```

**Atributos do Objeto****substituição de conexões de contabilidade**

Permite que os aplicativos substituam a configuração dos valores de contabilidade MQI e contabilidade de fila `values.This` é somente leitura.

**intervalo contábil**

Quanto tempo antes dos registros de conta intermediários serem gravados (em segundos). Esse atributo é somente leitura.

**gravação de atividade**

Controla a geração de relatório de atividades. Esse atributo é somente leitura.

**Adotar nova verificação de MCA**

Os elementos verificados para determinar se um MCA deve ser adotado quando um novo canal de entrada for detectado com o mesmo nome que um MCA que já está ativo. Esse atributo é somente leitura.

**Adotar novo tipo de MCA**

Se uma instância órfã de um MCA de um tipo de canal específico deve ser reiniciada automaticamente quando uma nova solicitação de canal de entrada correspondente a adotar novos parâmetros de verificação de mca for detectada Esse atributo é somente leitura.

**tipo de autenticação**

Indica o tipo da autenticação que está sendo executada

**Evento de Autoridade**

Controla os eventos de autoridade Esse atributo é somente leitura.

**Iniciar Opções**

Opções que se aplicam ao método `begin`. O valor inicial é MQBO\_NONE.



**evento de ponte**

Se eventos de ponte IMS são gerados. Esse atributo é somente leitura.

**Definição automática de canal**

Valor de definição automática do canal. Esse atributo é somente leitura.

**evento de definição automática do canal**

Valor do evento de definição automática do canal. Esse atributo é somente leitura.

**Saída de auto-definição de canal**

Nome da saída de definição automática do canal. Esse atributo é somente leitura.

**evento do canal**

Se os eventos do canal são gerados. Esse atributo é somente leitura.

**Adaptadores do iniciador de canais**

O número de subtarefas do adaptador a serem usadas para processar chamadas IBM MQ . Esse atributo é somente leitura.

**Controle de inicializador de canais**

Se o Inicializador de Canais deve ser iniciado automaticamente quando o Gerenciador de Filas for iniciado Esse atributo é somente leitura.

**Dispatchers do iniciador de canais**

O número de dispatchers para usar para o inicializador de canais. Esse atributo é somente leitura.

**autoinicialização de rastreamento do inicializador de canais**

Se o rastreamento do inicializador de canal deve ser iniciado automaticamente ou não. Esse atributo é somente leitura.

**Tamanho da tabela de rastreamento do iniciador de canais**

O tamanho do espaço para dados de rastreamento do inicializador de canais (em MB) Esse atributo é somente leitura.

**Monitoramento de canal**

Controla a coleção de dados de monitoramento on-line para canais. Esse atributo é somente leitura.

**referência de canal**

Uma referência a uma definição de canal para uso durante a conexão do cliente. Enquanto conectado, este atributo pode ser configurado como nulo, mas não pode ser alterado para qualquer outro valor. O valor inicial é nulo.

**estatísticas do canal**

Controla a coleção de dados estatísticos para canais. Esse atributo é somente leitura.

**conjunto de caracteres**

Identificador do conjunto de caracteres codificados (CCSID). Esse atributo é somente leitura.

**Monitoramento do Emissor de Cluster**

Controla a coleta de dados de monitoramento on-line para canais emissores de cluster definidos automaticamente. Esse atributo é somente leitura.

**Estatísticas do Emissor de Cluster**

Controla a coleta de dados estatísticos para canais emissores de cluster definidos automaticamente. Esse atributo é somente leitura.

**Dados de carga de trabalho do cluster**

Dados de Saída de Carga de Trabalho do Cluster Esse atributo é somente leitura.

**saída da carga de trabalho do cluster**

Nome da saída de carga de trabalho do cluster Esse atributo é somente leitura.

**Comprimento de carga de trabalho do cluster**

Comprimento da carga de trabalho do cluster Esse atributo é somente leitura.

**mru de carga de trabalho do cluster**

Valor de canais usados mais recentemente da carga de trabalho do cluster. Esse atributo é somente leitura.

**Fila de uso de carga de trabalho do cluster**

A carga de trabalho do cluster usa o valor da fila. Esse atributo é somente leitura.

**evento de comando**

Se os eventos de comando são gerados Esse atributo é somente leitura.

**Nome da fila de entrada do comando**

Nome da fila de entrada do comando do sistema. Esse atributo é somente leitura.

**nível de comando**

Nível de comando suportado pelo gerenciador de filas.. Esse atributo é somente leitura.

**Controle de servidor de comandos**

Se o Servidor de Comandos deve ser iniciado automaticamente quando o Gerenciador de Filas for iniciado Esse atributo é somente leitura.

**Opções de Conexão**

Opções que se aplicam ao método connect. O valor inicial é MQCNO\_NONE. Os seguintes valores adicionais podem ser possíveis, dependendo da plataforma:

- MQCNO\_STANDARD\_BINDING
- MQCNO\_FASTPATH\_BINDING
- MQCNO\_HANDLE\_SHARE\_NONE
- MQCNO\_HANDLE\_SHARE\_BLOCK
- MQCNO\_HANDLE\_SHARE\_NO\_BLOCK
- MQCNO\_SERIALIZE\_CONN\_TAG\_Q\_MGR
- MQCNO\_SERIALIZE\_CONN\_TAG\_QSG
- MQCNO\_RESTRICT\_CONN\_TAG\_Q\_MGR
- MQCNO\_RESTRICT\_CONN\_TAG\_QSG

**ID de conexão**

Um identificador exclusivo que permite que o MQ identifique de forma confiável um aplicativo.

**status da conexão**

TRUE quando conectado ao gerenciador de filas.. Esse atributo é somente leitura.

**tag de conexão**

Uma tag a ser associada a uma conexão Esse atributo só pode ser configurado quando não estiver conectado O valor inicial é nulo.

**hardware de criptografia**

Detalhes de configuração para hardware de criptografia. Para conexões do cliente MQI MQ .

**nome da fila de devoluções**

O nome da fila de mensagens não entregues Esse atributo é somente leitura.

**Nome da Fila de Transmissão Padrão**

Nome da fila de transmissão padrão.. Esse atributo é somente leitura.

**listas de distribuição**

Capacidade do gerenciador de filas para suportar listas de distribuições

**grupo de DNS**

O nome do grupo ao qual o listener TCP que manipula transmissões de entrada para o grupo de filas compartilhadas deve se associar ao usar o suporte do Workload Manager Dynamic Domain Name Services. Esse atributo é somente leitura.

**dns wlm**

Se o listener TCP que manipula transmissões de entrada para o grupo de filas compartilhadas deve registrar com o Workload Manager for Dynamic Domain Name Services. Esse atributo é somente leitura.

**primeiro registro de autenticação**

O primeiro de um ou mais objetos do Registro de classe ImqAuthentication, em nenhuma ordem específica, na qual a referência de conexão de Registro ImqAuthenticationendereço esse objeto. Para conexões do cliente MQI MQ .

**primeiro objeto gerenciado**

O primeiro de um ou mais objetos da classe ImqObject, em nenhuma ordem específica, na qual a referência de conexão ImqObject aborda esse objeto. O valor inicial é zero.

**Evento de Inibição**

Controla os eventos de inibição Esse atributo é somente leitura.

**Versão do endereço IP**

Qual protocolo IP (IPv4 ou IPv6) usar para uma conexão de canal. Esse atributo é somente leitura.

**repositório de chaves**

Local do arquivo de banco de dados de chaves no qual chaves e certificados são armazenados. Para conexões IBM MQ MQI client .

**contagem de reconfiguração de chave**

O número de bytes não criptografados enviados e recebidos dentro de uma conversa TLS antes que a chave secreta seja renegociada Esse atributo se aplica apenas a conexões do cliente usando MQCONNX. Consulte também [ssl key reset count](#).

**Cronômetro do listener**

O intervalo de tempo (em segundos) entre as tentativas do IBM MQ de reiniciar o listener se tiver havido uma falha de APPC ou TCP/IP Esse atributo é somente leitura.

**Evento Local**

Controla os eventos locais Esse atributo é somente leitura.

**evento do criador de logs**

Controla se os eventos do log de recuperação são gerados Esse atributo é somente leitura.

**Nome do grupo de LUs**

O nome da LU genérica que o listener da LU 6.2 que manipula as transmissões de entrada para o grupo de filas compartilhadas deve usar Esse atributo é somente leitura.

**Nome de LU**

O nome da LU a usar para transmissões de LU 6.2 de saída. Esse atributo é somente leitura.

**lu62 sufixo do braço**

O sufixo do SYS1.PARMLIB membro APPCPMxx, que nomeia LUADD para este inicializador de canais. Esse atributo é somente leitura.

**lu62 canais**

O número máximo de canais que podem ser atuais ou clientes que podem ser conectados, que usam o protocolo de transmissão LU 6.2 . Esse atributo é somente leitura.

**máximo de canais ativos**

O número máximo de canais que podem estar ativos a qualquer momento. Esse atributo é somente leitura.

**Máximo de canais**

O número máximo de canais que podem ser atuais (incluindo canais de conexão do servidor com clientes conectados). Esse atributo é somente leitura.

**Identificadores Máximos**

O número máximo de identificadores Esse atributo é somente leitura.

**comprimento máximo da mensagem**

O comprimento máximo possível para qualquer mensagem em qualquer fila gerenciada por este gerenciador de fila Esse atributo é somente leitura.

**Prioridade máxima**

Prioridade máxima da mensagem.. Esse atributo é somente leitura.

**Máx. de mens. não confirm.**

Número máximo de mensagens não confirmadas dentro de uma unidade ou trabalho Esse atributo é somente leitura.

**Contabilidade do MQI**

Controla a coleção de dados da conta para dados do MQI. Esse atributo é somente leitura.

**Estatística de MQI**

Controla a coleção de informações de monitoramento de estatísticas para o gerenciador de filas. Esse atributo é somente leitura.

**máximo da porta de saída**

A extremidade superior do intervalo de números de porta a ser usado ao ligar os canais de saída. Esse atributo é somente leitura.

**mínimo da porta de saída**

A extremidade inferior do intervalo de números de portas a ser utilizado ao ligar canais de saída. Esse atributo é somente leitura.

**senha**

senha associada ao ID do usuário

**evento de desempenho**

Controla os eventos de desempenho. Esse atributo é somente leitura.

**platform**

Plataforma na qual o gerenciador de filas reside. Esse atributo é somente leitura.

**Enfileiramento de Contas**

Controla a coleção de dados da conta para filas. Esse atributo é somente leitura.

**Monitoramento de Fila**

Controla a coleção de dados de monitoramento on-line para filas. Esse atributo é somente leitura.

**estatísticas de fila**

Controla a coleção de dados estatísticos para filas. Esse atributo é somente leitura.

**Tempo limite de recebimento**

Aproximadamente quanto tempo um canal de mensagem TCP/IP aguardará para receber dados, incluindo pulsações, de seu parceiro, antes de retornar ao estado inativo. Esse atributo é somente leitura.

**mínimo de tempo limite de recebimento**

O tempo mínimo que um canal TCP/IP aguardará para receber dados, incluindo pulsações, de seu parceiro, antes de retornar ao estado inativo. Esse atributo é somente leitura.

**Tipo de tempo limite de recebimento**

Um qualificador aplicado para receber tempo limite. Esse atributo é somente leitura.

**Evento Remoto**

Controla os eventos remotos. Esse atributo é somente leitura.

**nome de repositório**

O nome do repositório. Esse atributo é somente leitura.

**Lista de Nomes do Repositório**

Nome da lista de nomes de repositórios. Esse atributo é somente leitura.

**nome do gerenciador de filas compartilhadas**

Se MQOPENS de uma fila compartilhada em que o Nome do ObjectQMgré outro gerenciador de filas no grupo de filas compartilhadas devem ser resolvidos para uma abertura da fila compartilhada no gerenciador de fila local... Esse atributo é somente leitura.

**evento ssl**

Se eventos de SSL são gerados. Esse atributo é somente leitura.

**SSL FIPS obrigatório**

Se apenas os algoritmos certificados pelo FIPS devem ser usados se a criptografia for executada no software IBM MQ. Esse atributo é somente leitura.

**Contagem de reconfigurações de chave SSL**

O número de bytes não criptografados enviados e recebidos em uma conversa SSL antes da chave secreta ser renegociada. Esse atributo é somente leitura.

**evento de parada inicial**


Controla eventos de parada inicial. Esse atributo é somente leitura.

### intervalo de estatísticas

A frequência com que os dados de monitoramento de estatísticas são gravados na fila de monitoramento. Esse atributo é somente leitura.

### Disponibilidade do ponto de sincronização

Disponibilidade de participação de ponto de sincronização. Esse atributo é somente leitura.

**Nota:** Unidades globais de trabalho coordenadas pelo gerenciador de filas não são suportadas na plataforma IBM i.  É possível programar uma unidade de trabalho, coordenada externamente por IBM i, usando as chamadas do sistema nativo `_Rcommit` e `_Rback`. Inicie esse tipo de unidade de trabalho iniciando o aplicativo IBM MQ sob o controle de confirmação de nível de tarefa utilizando o comando `STRCMTCTL`. Consulte [Interfaces com o IBM i gerenciador de ponto de sincronização externo](#), para obter detalhes adicionais. A restauração e a confirmação são suportadas na plataforma IBM i para unidades locais de trabalho coordenadas por um gerenciador de filas.

### canais tcp

O número máximo de canais que podem ser atuais ou clientes que podem ser conectados que usam o protocolo de transmissão TCP/IP. Esse atributo é somente leitura.

### Manutenção de atividade TCP

Se o recurso TCP KEEPALIVE será usado para verificar se a outra extremidade da conexão ainda está disponível. Esse atributo é somente leitura.

### Nome do TCP

O nome do sistema TCP/IP único ou padrão a ser usado, dependendo do valor do tipo de pilha de tcp. Esse atributo é somente leitura.

### Tipo de pilha TCP

Se o inicializador de canais tem permissão para usar apenas o espaço de endereço TCP/IP especificado no nome tcp ou pode ligar a qualquer endereço TCP/IP selecionado. Esse atributo é somente leitura.

### Registro de Rota de Rastreo

Controla a gravação das informações de rastreo da rota. Esse atributo é somente leitura.

### Intervalo de ativação

Intervalo do acionador. Esse atributo é somente leitura.

### ID do usuário

Em plataformas AIX and Linux, o ID do usuário real do aplicativo. Nas plataformas Windows, o ID do usuário do aplicativo.

## Construtores

### **ImqQueueManager ();**

O construtor padrão..

### **ImqQueueManager (const ImqQueueManager & manager );**

O construtor de cópia.. O status da conexão será FALSE

### **ImqQueueManager (const char \* name );**

Configura o nome de ImqObject para *name*

## Destruidores

Quando um objeto ImqQueueManager é destruído, ele é automaticamente desconectado.

## Métodos de classe (público).

### **comportamento MQLONG estático ();**

Retorna o comportamento..

### **void setBehavior(const MQLONG behavior = 0);**

Configura o comportamento.

## Métodos de objeto (público)

### **operador void = (const ImqQueueManager & mgr );**

Desconecta se necessário e copia dados de instância de *mgr*. O status da conexão é FALSE

### **ImqBoolean accountingConnSubstituir (MQLONG & statint);**

Fornece uma cópia do valor de substituição de conexões de contabilidade. Ele retornará TRUE se for bem-sucedido

### **MQLONG accountingConnOverride ();**

Retorna o valor de substituição de conexões de contabilidade sem qualquer indicação de possíveis erros.

### **ImqBoolean accountingInterval (MQLONG & statint);**

Fornece uma cópia do valor do intervalo contábil Ele retornará TRUE se for bem-sucedido

### **MQLONG accountingInterval ();**

Retorna o valor do intervalo contábil sem qualquer indicação de possíveis erros

### **ImqBoolean activityRecording (MQLONG & rec);**

Fornece uma cópia do valor de registro da atividade Ele retornará TRUE se for bem-sucedido

### **MQLONG activityRecording ();**

Retorna o valor de gravação da atividade sem qualquer indicação de possíveis erros...

### **ImqBoolean adoptNewMCACheck (MQLONG & check);**

Fornece uma cópia do novo valor de verificação do MCA. Ele retornará TRUE se for bem-sucedido

### **MQLONG adoptNewMCACheck ();**

Retorna o novo valor de verificação do MCA sem qualquer indicação de possíveis erros.

### **ImqBoolean adoptNewMCAType (MQLONG & type);**

Fornece uma cópia do tipo adotar novo MCA. Ele retornará TRUE se for bem-sucedido

### **MQLONG adoptNewMCAType ();**

Retorna o novo tipo de MCA sem qualquer indicação de possíveis erros.

### **QLONG authenticationType () const;**

Retorna o tipo de autenticação..

### **void setAuthenticationType (const MQLONG type = MQCSP\_AUTH\_NONE);**

Configura o tipo de autenticação

### **ImqBoolean authorityEvent(MQLONG & evento );**

Fornece uma cópia de um estado de ativação do evento de autoridade Ele retornará TRUE se for bem-sucedido

### **MQLONG authorityEvent();**

Retorna o estado de ativação do evento de autoridade sem qualquer indicação de possíveis erros..

### **ImqBoolean backout ();**

Restaura as mudanças não confirmados Ele retornará TRUE se for bem-sucedido

### **ImqBoolean begin ();**

Inicia uma unidade de serviço O iniciar opções afetam o comportamento deste método. Ele retornará TRUE se for bem-sucedido, mas também retornará TRUE mesmo se a chamada MQBEGIN subjacente retornar MQRC\_NO\_EXTERNAL\_PARTICIPANTES ou MQRC\_PARTICIPANT\_NOT\_AVAILABLE (ambos associados a MQCC\_WARNING).

### **MQLONG beginOptions() const;**

Retorna as opções de início.

### **void setBeginOptions (const MQLONG options = MQBO\_NONE);**

Configura as opções de início

### **ImqBoolean bridgeEvent (MQLONG & event);**

Fornece uma cópia do valor do evento de ponte Ele retornará TRUE se for bem-sucedido

### **MQLONG bridgeEvent ();**

Retorna o valor do evento de ponte sem qualquer indicação de possíveis erros..

**ImqBoolean channelAutoDefinição (MQLONG & value );**

Fornecer uma cópia do valor de definição automática do canal.. Ele retornará TRUE se for bem-sucedido

**MQLONG channelAutoDefinição ();**

Retorna o valor de definição automática do canal sem qualquer indicação de possíveis erros

**ImqBoolean channelAutoDefinitionEvent(valor MQLONG & );**

Fornecer uma cópia do valor do evento de definição automática do canal Ele retornará TRUE se for bem-sucedido

**MQLONG channelAutoDefinitionEvent();**

Retorna o valor do evento de definição automática do canal sem qualquer indicação de possíveis erros..

**ImqBoolean channelAutoDefinitionExit( ImqString & name );**

Fornecer uma cópia do nome de saída de definição automática do canal.. Ele retornará TRUE se for bem-sucedido

**ImqString channelAutoDefinitionExit( );**

Retorna o nome da saída de definição automática do canal sem qualquer indicação de possíveis erros

**ImqBoolean channelEvent (MQLONG & event);**

Fornecer uma cópia do valor do evento do canal Ele retornará TRUE se for bem-sucedido

**MQLONG channelEvent();**

Retorna o valor do evento do canal sem nenhuma indicação de possíveis erros..

**MQLONG channelInitiatorAdapters ();**

Retorna o valor dos adaptadores do inicializador de canais sem qualquer indicação de possíveis erros..

**Adaptadores ImqBoolean channelInitiator(MQLONG & adaptadores);**

Fornecer uma cópia do valor dos adaptadores do inicializador de canais Ele retornará TRUE se for bem-sucedido

**MQLONG channelInitiatorControl ();**

Retorna o valor de inicialização do inicializador de canais sem qualquer indicação de possíveis erros

**Controle ImqBoolean channelInitiator(MQLONG & init);**

Fornecer uma cópia do valor de inicialização do controle do inicializador de canais Ele retornará TRUE se for bem-sucedido

**MQLONG channelInitiatorDispatchers ();**

Retorna o valor de dispatchers do inicializador de canais sem qualquer indicação de possíveis erros..

**ImqBoolean channelInitiatorDispatchers (MQLONG e dispatchers)**

Fornecer uma cópia do valor dos dispatchers do inicializador de canais Ele retornará TRUE se for bem-sucedido

**MQLONG channelInitiatorTraceAutoStart ();**

Retorna o valor de início automático de rastreamento do inicializador de canais sem qualquer indicação de possíveis erros..

**ImqBoolean channelInitiatorTraceAutoIniciar (MQLONG & auto).**

Fornecer uma cópia do valor de início automático do rastreamento do inicializador de canais Ele retornará TRUE se for bem-sucedido

**MQLONG channelInitiatorTraceTableTamanho ();**

Retorna o valor do tamanho da tabela de rastreamento do inicializador de canais sem qualquer indicação de possíveis erros.

**ImqBoolean channelInitiatorTraceTableTamanho (MQLONG & size);**

Fornecer uma cópia do valor de tamanho da tabela de rastreamento do inicializador de canais. Ele retornará TRUE se for bem-sucedido

**ImqBoolean channelMonitoring (MQLONG & monchl);**

Fornecer uma cópia do valor de monitoramento do canal. Ele retornará TRUE se for bem-sucedido

**MQLONG channelMonitoring (); e**

Retorna o valor de monitoramento do canal sem qualquer indicação de possíveis erros..

**ImqBoolean channelReference( ImqChannel \* & pchannel );**

Fornece uma cópia da referência de canal.. Se a referência de canal for inválida, configure *pchannel* como nulo. Esse método retornará TRUE se for bem-sucedido

**ImqChannel \* channelReference( );**

Retorna a referência de canal sem qualquer indicação de possíveis erros

**ImqBoolean setChannelReferência ( ImqChannel & canal )**

Configura a referência de canal Esse método retornará TRUE se for bem-sucedido

**Referência ImqBoolean setChannel( ImqChannel \* channel = 0);**

Configura ou reconfigura a referência de canal.. Esse método retornará TRUE se for bem-sucedido

**ImqBoolean channelStatistics (MQLONG e statchl);**

Fornece uma cópia do valor de estatísticas do canal Ele retornará TRUE se for bem-sucedido

**MQLONG channelStatistics ();**

Retorna o valor de estatísticas do canal sem qualquer indicação de possíveis erros...

**ImqBoolean characterSet(MQLONG & ccsid );**

Fornece uma cópia de um conjunto de caracteres Ele retornará TRUE se for bem-sucedido

**MQLONG characterSet();**

Retorna uma cópia do conjunto de caracteres, sem qualquer indicação de possíveis erros

**MQLONG clientSslKeyResetCount () const;**

Retorna o valor de contagem de reconfiguração de chave SSL usado em conexões do cliente...

**void setClientSslKeyResetCount(contagem const MQLONG);**

Configura a contagem de reconfiguração de chave SSL usada em conexões do cliente

**ImqBoolean clusterSenderMonitoring (MQLONG & monacIs);**

Fornece uma cópia do valor padrão de monitoramento do emissor de cluster. Ele retornará TRUE se for bem-sucedido

**MQLONG clusterSenderMonitoring ();**

Retorna o valor padrão de monitoramento do emissor de cluster sem qualquer indicação de possíveis erros..

**ImqBoolean clusterSenderImqBoolean (MQLONG & statacls);**

Fornece uma cópia do valor de estatísticas do emissor de clusters Ele retornará TRUE se for bem-sucedido

**MQLONG clusterSenderStatistics ();**

Retorna o valor de estatísticas do emissor do cluster sem qualquer indicação de possíveis erros

**ImqBoolean clusterWorkloadDados ( ImqString & dados );**

Fornece uma cópia dos dados de saída de carga do cluster. Ele retornará TRUE se for bem-sucedido

**ImqString clusterWorkloadDados ();**

Retorna os dados de saída de carga de trabalho do cluster sem qualquer indicação de possíveis erros

**ImqBoolean clusterWorkloadExit ( ImqString & name );**

Fornece uma cópia do nome da saída de carga do cluster. Ele retornará TRUE se for bem-sucedido

**ImqString clusterWorkloadSaída ();**

Retorna o nome de saída da carga de trabalho do cluster sem qualquer indicação de possíveis erros

**ImqBoolean clusterWorkloadComprimento (MQLONG & length );**

Fornece uma cópia da duração da carga de trabalho do cluster Ele retornará TRUE se for bem-sucedido

**MQLONG clusterWorkloadLength ();**

Retorna o comprimento da carga de trabalho do cluster sem qualquer indicação de possíveis erros

**ImqBoolean clusterWorkLoadMRU (MQLONG & mru)**

Fornece uma cópia do valor de canais usados mais recentemente da carga de trabalho do cluster. Ele retornará TRUE se for bem-sucedido



**MQLONG clusterWorkLoadMRU ();**

Retorna o valor dos canais usados mais recentemente da carga de trabalho do cluster sem qualquer indicação de possíveis erros

**ImqBoolean clusterWorkLoadUseQ (MQLONG & useq);**

Fornece uma cópia do valor da fila de uso da carga de trabalho do cluster Ele retornará TRUE se for bem-sucedido

**MQLONG clusterWorkLoadUseQ ();**

Retorna o valor da fila de uso da carga de trabalho do cluster sem nenhuma indicação de possíveis erros..

**ImqBoolean commandEvent (MQLONG & event);**

Fornece uma cópia do valor de evento de comando. Ele retornará TRUE se for bem-sucedido

**MQLONG commandEvent ();**

Retorna o valor do evento de comando sem qualquer indicação de possíveis erros..

**ImqBoolean commandInputQueueName( ImqString & name );**

Fornece uma cópia do nome da fila de entrada de comandos Ele retornará TRUE se for bem-sucedido

**ImqString commandInputQueueName( );**

Retorna o nome da fila de entrada de comandos sem qualquer indicação de possíveis erros

**ImqBoolean commandLevel(MQLONG & level );**

Fornece uma cópia do nível de comando.. Ele retornará TRUE se for bem-sucedido

**MQLONG commandLevel();**

Retorna o nível de comando sem qualquer indicação de possíveis erros

**MQLONG commandServerControl ();**

Retorna o valor de inicialização do servidor de comando sem qualquer indicação de possíveis erros...

**ImqBoolean commandServerControl (MQLONG & servidor);**

Fornece uma cópia de um valor de inicialização de controle do servidor de comando Ele retornará TRUE se for bem-sucedido

**ImqBoolean commit ();**

Confirma as mudanças não confirmados Ele retornará TRUE se for bem-sucedido

**ImqBoolean connect ();**

Conecta ao gerenciador de filas com o nome ImqObject fornecido, o padrão sendo o gerenciador de filas locais. Se desejar conectar-se a um gerenciador de filas específico, use o método ImqObject setName antes da conexão Se houver uma referência de canal, ela será usada para transmitir informações sobre a definição do canal para MQCONN em um MQCD.. O ChannelType no MQCD é configurado como MQCHT\_CLNTCONN. informações de referência de canal, que são apenas significativas para conexões do cliente, são ignoradas para conexões do servidor. As opções de conexão afetam o comportamento desse método.. Este método configura o status da conexão para TRUE se for bem-sucedido Ele retorna o novo status da conexão

Se houver um primeiro registro de autenticação, a cadeia de registros de autenticação será usada para autenticar certificados digitais para os canais do cliente seguros

É possível conectar mais de um objeto do Gerenciador de ImqQueue ao mesmo gerenciador de fila Todos usam o mesmo identificador de conexão MQHCONN e compartilham a funcionalidade de UOW para a conexão associada ao encadeamento O primeiro gerenciador de ImqQueue a se conectar obtém o identificador MQHCONN O último ImqQueueManager para desconectar executa o MQDISC.

Para um programa multiencadeado, é recomendado que um objeto ImqQueueManager separado seja usado para cada encadeamento.

**ImqBinary connectionId () const;**

Retorna o ID da conexão.

**ImqBinary connectionTag () const;**

Retorna a tag de conexão

**ImqBoolean setConnectionTag (const MQBYTE128 tag = 0);**  
 Configura a tag de conexão Se *tag* for zero, limpa a tag de conexão. Esse método retornará TRUE se for bem-sucedido

**ImqBoolean setConnectionTag (const ImqBinary & tag );**  
 Configura a tag de conexão O comprimento dos dados da *tag* deve ser zero (para limpar a tag de conexão) ou MQ\_CONN\_TAG\_LENGTH.. Esse método retornará TRUE se for bem-sucedido

**MQLONG connectOptions() const;**  
 Retorna as opções de conexão..

**void setConnectOptions (const MQLONG options = MQCNO\_NONE);**  
 Configura as opções de conexão

**ImqBoolean connectionStatus() const;**  
 Retorna o status da conexão..

**ImqString cryptographicHardware ( );**  
 Retorna o hardware criptográfico

**ImqBoolean setCryptographicHardware (const char \* hardware = 0);**  
 Configura o hardware criptográfico Esse método retornará TRUE se for bem-sucedido

**ImqBoolean deadLetterQueueName( ImqString & name );**  
 Fornece uma cópia de um nome da fila de mensagens não entregues. Ele retornará TRUE se for bem-sucedido

**ImqString deadLetterQueueName( );**  
 Retorna uma cópia do nome da fila de devoluções, sem qualquer indicação de possíveis erros

**ImqBoolean defaultTransmissionQueueName( ImqString & name );**  
 Fornece uma cópia da fila de transmissão padrão. Ele retornará TRUE se for bem-sucedido

**ImqString defaultTransmissionQueueName( );**  
 Retorna o nome da fila de transmissão padrão sem qualquer indicação de possíveis erros

**ImqBoolean disconnect ( );**  
 Desconecta do gerenciador de filas e configura o status da conexão para FALSE. Fecha todos os objetos ImqProcess e ImqQueue associados a esse objeto e corta suas referências de conexão antes da desconexão Se mais de um objeto do Gerenciador de ImqQueue estiver conectado ao mesmo gerenciador de filas, apenas o último a se desconectar executará uma desconexão física; outros executarão uma desconexão lógica As mudanças não confirmadas são confirmadas apenas na desconexão física

Esse método retornará TRUE se for bem-sucedido Se for chamado quando não houver conexão existente, o código de retorno também será true.

**ImqBoolean distributionLists(MQLONG & support );**  
 Fornece uma cópia do valor das listas de distribuição. Ele retornará TRUE se for bem-sucedido

**MQLONG distributionLists();**  
 Retorna o valor da lista de distribuições sem qualquer indicação de possíveis erros

**ImqBoolean dnsGroup ( ImqString & group);**  
 Fornece uma cópia do nome do grupo de DNS. Ele retornará TRUE se for bem-sucedido

**ImqString dnsGroup ( );**  
 Retorna o nome do grupo de DNS sem qualquer indicação de possíveis erros

**ImqBoolean dnsWlm (MQLONG & wlm);**  
 Fornece uma cópia do valor do DNS WLM. Ele retornará TRUE se for bem-sucedido

**MQLONG dnsWlm ( );**  
 Retorna o valor do WLM do DNS sem qualquer indicação de possíveis erros

**ImqAuthenticationRegistro \* firstAuthenticationRegistro ( ) const;**  
 Retorna o primeiro registro de autenticação..

**void setFirstAuthenticationRecord (const ImqAuthenticationRecord \* air = 0);**  
 Configura o primeiro registro de autenticação

**ImqObject \* firstManagedObject () const;**

Retorna o primeiro objeto gerenciado..

**ImqBoolean inhibitEvent(MQLONG & event );**

Fornece uma cópia de um estado de ativação do evento de inibição Ele retornará TRUE se for bem-sucedido

**MQLONG inhibitEvent();**

Retorna o estado de ativação do evento de inibição sem qualquer indicação de possíveis erros..

**ImqBoolean ipAddressVersão (MQLONG & version);**

Fornece uma cópia do valor da versão do endereço IP Ele retornará TRUE se for bem-sucedido

**MQLONG ipAddressVersão ();**

Retorna o valor da versão do endereço IP sem qualquer indicação de possíveis erros..

**ImqBoolean keepAlive (MQLONG & keepalive)**

Fornece uma cópia do valor manter ativo. Ele retornará TRUE se for bem-sucedido

**MQLONG keepAlive ();**

Retorna o valor keep-alive sem qualquer indicação de possíveis erros

**ImqString keyRepository ( );**

Retorna o repositório de chaves.

**ImqBoolean setKeyRepository (const char \* repository = 0);**

Configura o repositório de chaves.. Ele retornará TRUE se for bem-sucedido

**ImqBoolean listenerTimer (MQLONG & timer);**

Fornece uma cópia do valor do cronômetro do listener. Ele retornará TRUE se for bem-sucedido

**MQLONG listenerTimer ();**

Retorna o valor do cronômetro do listener sem qualquer indicação de erros possíveis

**ImqBoolean localEvent(MQLONG & evento );**

Fornece uma cópia do estado de ativação do evento local.. Ele retornará TRUE se for bem-sucedido

**MQLONG localEvent();**

Retorna o estado de ativação do evento local sem qualquer indicação de possíveis erros..

**ImqBoolean loggerEvent (MQLONG & count)**

Fornece uma cópia do valor do evento do criador de logs Ele retornará TRUE se for bem-sucedido

**MQLONG loggerEvent ();**

Retorna o valor do evento do criador de logs sem qualquer indicação de possíveis erros..

**ImqBoolean luGroupName ( ImqString & name);**

Fornece uma cópia do nome do grupo de LUs Ele retorna TRUE se for bem-sucedido

**ImqString luGroupName ();**

Retorna o nome do grupo de LUs sem qualquer indicação de possíveis erros

**ImqBoolean lu62ARMSuffix ( ImqString & sufixo)**

Fornece uma cópia do sufixo do ARM LU62 Ele retornará TRUE se for bem-sucedido

**ImqString lu62ARMSuffix ( );**

Retorna o sufixo ARM LU62 sem qualquer indicação de possíveis erros

**ImqBoolean luName ( ImqString & name);**

Fornece uma cópia do nome da LU.. Ele retornará TRUE se for bem-sucedido

**ImqString luName ( );**

Retorna o nome da LU sem qualquer indicação de possíveis erros

**ImqBoolean maximumActiveCanais (MQLONG & canais).**

Fornece uma cópia do valor do máximo de canais ativos. Ele retornará TRUE se for bem-sucedido

**MQLONG maximumActiveChannels ();**

Retorna o valor do máximo de canais ativos sem qualquer indicação de possíveis erros.

**ImqBoolean maximumCurrentCanais (MQLONG & canais);**

Fornece uma cópia do valor máximo de canais atuais. Ele retornará TRUE se for bem-sucedido

**MQLONG maximumCurrentCanais ();**

Retorna o valor máximo de canais atuais sem qualquer indicação de possíveis erros..

**ImqBoolean maximumHandles(MQLONG & number );**

Fornece uma cópia do máximo de identificadores. Ele retornará TRUE se for bem-sucedido

**MQLONG maximumHandles();**

Retorna o máximo de identificadores sem qualquer indicação de possíveis erros

**ImqBoolean maximumLu62Channels (MQLONG & canais);**

Fornece uma cópia do valor máximo de LU62 de canais. Ele retornará TRUE se for bem-sucedido

**MQLONG maximumLu62Channels ();**

Retorna o valor máximo de LU62 canais sem qualquer indicação de possíveis erros

**ImqBoolean maximumMessageLength (MQLONG & length );**

Fornece uma cópia do comprimento máximo da mensagem.. Ele retornará TRUE se for bem-sucedido

**MQLONG maximumMessageLength ();**

Retorna o comprimento máximo da mensagem sem qualquer indicação de possíveis erros..

**ImqBoolean maximumPriority(MQLONG & priority );**

Fornece uma cópia da prioridade máxima. Ele retornará TRUE se for bem-sucedido

**MQLONG maximumPriority();**

Retorna uma cópia da prioridade máxima, sem qualquer indicação de possíveis erros

**ImqBoolean maximumTcp(MQLONG e canais);**

Fornece uma cópia do valor máximo de canais TCP. Ele retornará TRUE se for bem-sucedido

**MQLONG maximumTcpChannels ();**

Retorna o valor máximo de canais TCP sem qualquer indicação de possíveis erros..

**ImqBoolean maximumUncommittedMensagens (MQLONG & number )**

Fornece uma cópia do número máximo de mensagens não confirmados Ele retornará TRUE se for bem-sucedido

**MQLONG maximumUncommittedMessages ();**

Retorna o máximo de mensagens não consolidadas sem qualquer indicação de possíveis erros.

**ImqBoolean mqiAccounting (MQLONG & statint);**

Fornece uma cópia do valor de contabilidade MQI. Ele retornará TRUE se for bem-sucedido

**MQLONG mqiAccounting ();**

Retorna o valor de contabilidade MQI sem qualquer indicação de possíveis erros.

**ImqBoolean mqiStatistics (MQLONG & statmqi)**

Fornece uma cópia do valor de estatísticas de MQI. Ele retornará TRUE se for bem-sucedido

**MQLONG mqiStatistics ();**

Retorna o valor de estatísticas de MQI sem qualquer indicação de erros possíveis.

**ImqBoolean outboundPortMáx. (MQLONG & max);**

Fornece uma cópia do valor máximo da porta de saída.. Ele retornará TRUE se for bem-sucedido

**MQLONG outboundPortMax ();**

Retorna o valor máximo da porta de saída sem qualquer indicação de possíveis erros..

**ImqBoolean outboundPortMin (MQLONG & min);**

Fornece uma cópia do valor mínimo da porta de saída Ele retornará TRUE se for bem-sucedido

**MQLONG outboundPortMin ();**

Retorna o valor mínimo da porta de saída sem qualquer indicação de possíveis erros...

**ImqBinary senha () const;**

Retorna a senha usada em conexões do cliente

**ImqBoolean setPassword (const ImqString & password);**

Configura a senha usada nas conexões do cliente

**ImqBoolean setPassword (const char \* = 0 senha);**

Configura a senha usada nas conexões do cliente

**ImqBoolean setPassword (const ImqBinary & password);**

Configura a senha usada nas conexões do cliente

**ImqBoolean performanceEvent(MQLONG & evento );**

Fornece uma cópia do estado de ativação do evento de desempenho.. Ele retornará TRUE se for bem-sucedido

**MQLONG performanceEvent();**

Retorna o estado de ativação do evento de desempenho sem qualquer indicação de possíveis erros..

**Plataforma ImqBoolean (plataforma MQLONG & );**

Fornece uma cópia de uma plataforma. Ele retornará TRUE se for bem-sucedido

**Plataforma MQLONG ();**

Retorna a plataforma sem qualquer indicação de possíveis erros..

**ImqBoolean queueAccounting (MQLONG & acctq)**

Fornece uma cópia do valor de contabilidade da fila. Ele retornará TRUE se for bem-sucedido

**MQLONG queueAccounting ();**

Retorna o valor contábil da fila sem qualquer indicação de possíveis erros

**ImqBoolean queueMonitoring (MQLONG & monq);**

Fornece uma cópia do valor de monitoramento de fila. Ele retornará TRUE se for bem-sucedido

**MQLONG queueMonitoring ();**

Retorna o valor de monitoramento da fila sem qualquer indicação de erros possíveis

**ImqBoolean queueStatistics (MQLONG & statq)**

Fornece uma cópia do valor de estatísticas de fila. Ele retornará TRUE se for bem-sucedido

**MQLONG queueStatistics ();**

Retorna o valor de estatísticas de fila sem qualquer indicação de possíveis erros..

**ImqBoolean receiveTimeout (MQLONG & timeout);**

Fornece uma cópia do valor de tempo limite de recebimento Ele retornará TRUE se for bem-sucedido

**MQLONG receiveTimeout ();**

Retorna o valor de tempo limite de recebimento sem qualquer indicação de erros possíveis..

**ImqBoolean receiveTimeoutMin (MQLONG & min);**

Fornece uma cópia do valor de tempo limite mínimo de recebimento Ele retornará TRUE se for bem-sucedido

**MQLONG receiveTimeoutMin ();**

Retorna o valor mínimo de tempo limite de recebimento sem qualquer indicação de possíveis erros..

**ImqBoolean receiveTimeoutTipo (MQLONG & type);**

Fornece uma cópia do tipo de tempo limite de recebimento.. Ele retornará TRUE se for bem-sucedido

**MQLONG receiveTimeoutType ();**

Retorna o tipo de tempo limite de recebimento sem qualquer indicação de possíveis erros

**ImqBoolean remoteEvent(MQLONG & event );**

Fornece uma cópia de um estado de ativação do evento remoto Ele retornará TRUE se for bem-sucedido

**MQLONG remoteEvent();**

Retorna o estado de ativação do evento remoto sem qualquer indicação de possíveis erros

**ImqBoolean repositoryName( ImqString & name );**

Fornece uma cópia de um nome de repositório. Ele retornará TRUE se for bem-sucedido

**ImqString repositoryName( );**

Retorna o nome do repositório sem qualquer indicação de possíveis erros

**ImqBoolean repositoryNameListNome ( ImqString & nome );**

Fornece uma cópia do nome da lista de nomes de repositórios Ele retornará TRUE se for bem-sucedido

**ImqString repositoryNameListNome ();**

Retorna uma cópia do nome da lista de nomes de repositórios sem qualquer indicação de possíveis erros

**ImqBoolean sharedQueueQueueManagerNome (MQLONG & name);;**

Fornecer uma cópia do valor do nome do gerenciador de filas compartilhadas. Ele retornará TRUE se for bem-sucedido

**MQLONG sharedQueueQueueManagerName ();**

Retorna o valor do nome do gerenciador de filas compartilhadas sem qualquer indicação de possíveis erros.

**ImqBoolean sslEvent (MQLONG & evento);**

Fornecer uma cópia do valor do evento SSL. Ele retornará TRUE se for bem-sucedido

**MQLONG sslEvent ();**

Retorna o valor do evento SSL sem qualquer indicação de possíveis erros..

**ImqBoolean sslFips (MQLONG & sslfips).**

Fornecer uma cópia do valor SSL FIPS. Ele retornará TRUE se for bem-sucedido

**MQLONG sslFips ();**

Retorna o valor FIPS SSL sem qualquer indicação de possíveis erros.

**ImqBoolean sslKeyResetCount (MQLONG & count);**

Fornecer uma cópia do valor de contagem de reconfiguração de chave SSL. Ele retornará TRUE se for bem-sucedido

**MQLONG sslKeyResetCount ();**

Retorna o valor de contagem de reconfiguração de chave SSL sem qualquer indicação de possíveis erros

**Evento ImqBoolean startStop(evento MQLONG & );**

Fornecer uma cópia do estado de ativação do evento de inicialização. Ele retornará TRUE se for bem-sucedido

**MQLONG startStopEvent ();**

Retorna o estado de ativação do evento de inicialização sem qualquer indicação de possíveis erros.

**ImqBoolean statisticsInterval (MQLONG & statint);**

Fornecer uma cópia do valor do intervalo de estatísticas. Ele retornará TRUE se for bem-sucedido

**MQLONG statisticsInterval ();**

Retorna o valor do intervalo de estatísticas sem qualquer indicação de possíveis erros..

**ImqBoolean syncPointDisponibilidade (MQLONG & sync );**

Fornecer uma cópia do valor de disponibilidade do ponto de sincronização. Ele retornará TRUE se for bem-sucedido

**MQLONG syncPointAvailability ();**

Retorna uma cópia do valor de disponibilidade do ponto de sincronização, sem qualquer indicação de possíveis erros

**ImqBoolean tcpName ( ImqString & name);**

Fornecer uma cópia de um nome do sistema TCP. Ele retornará TRUE se for bem-sucedido

**ImqString tcpName ();**

Retorna o nome do sistema TCP sem qualquer indicação de possíveis erros

**ImqBoolean tcpStackTipo (MQLONG & type);**

Fornecer uma cópia do tipo de pilha TCP.. Ele retornará TRUE se for bem-sucedido

**MQLONG tcpStackType ();**

Retorna o tipo de pilha TCP sem qualquer indicação de possíveis erros..

**Gravação de ImqBoolean traceRoute(MQLONG & routerec)**

Fornecer uma cópia do valor de registro da rota de rastreamento. Ele retornará TRUE se for bem-sucedido

**MQLONG traceRouteRecording ();**

Retorna o valor de gravação de rota de rastreamento sem qualquer indicação de possíveis erros..

**ImqBoolean triggerInterval(MQLONG & interval );**

Fornecer uma cópia do intervalo acionador. Ele retornará TRUE se for bem-sucedido

**MQLONG triggerInterval();**

Retorna o intervalo do acionador sem qualquer indicação de erros possíveis..

**ImqBinary userId () const;**

Retorna o ID do usuário usado nas conexões do cliente

**ImqBoolean setUserId (const ImqString & id);**

Configura o ID do usuário usado em conexões do cliente

**ImqBoolean setUserId (const char \* = 0 id);**

Configura o ID do usuário usado em conexões do cliente

**ImqBoolean setUserId (const ImqBinary & id);**

Configura o ID do usuário usado em conexões do cliente

**Métodos de objeto (protegidos)****void setFirstManagedObject (const ImqObject \* object = 0);**

Configura o primeiro objeto gerenciado

**Dados do objeto (protegidos).****MQHCONN ohconn**

O identificador de conexão IBM MQ (significativo apenas enquanto o status da conexão for TRUE).

**Códigos de Razão**

- MQRC\_ATTRIBUTE\_LOCKED
- MQRC\_ENVIRONMENT\_ERROR
- MQRC\_FUNCTION\_NOT\_SUPPORTED
- MQRC\_REFERENCE\_ERROR
- (códigos de razão para MQBACK)
- (códigos de razão para MQBEGIN)
- (códigos de razão para MQCMIT)
- (códigos de razão para MQCONN)
- (códigos de razão para MQDISC)
- (códigos de razão para MQCONN)

**Classe C++ do cabeçalho ImqReference**

Esta classe contém recursos da estrutura de dados MQRMH.

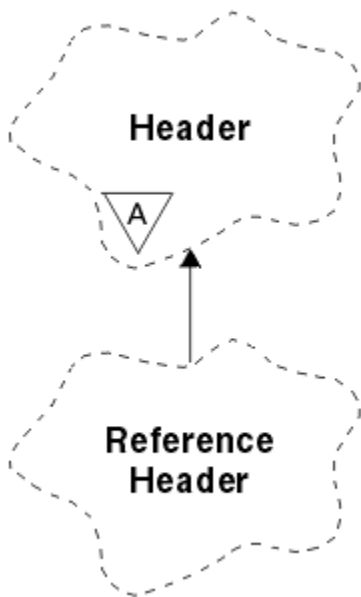


Figura 34. Classe de cabeçalho *ImqReference*

Essa classe está relacionada às chamadas MQI listadas em [“Referência cruzada do cabeçalho ImqReference”](#) na página 1864.

- [“Atributos do Objeto”](#) na página 1948
- [“Construtores”](#) na página 1949
- [“Métodos ImqItem sobrecarregados”](#) na página 1949
- [“Métodos de objeto \(público\)”](#) na página 1949
- [“Dados do objeto \(protegidos\).”](#) na página 1950
- [“Códigos de Razão”](#) na página 1950

## Atributos do Objeto

### ambiente de destino

Ambiente do destino. O valor inicial é uma cadeia nula.

### nome do destino

Nome do destino de dados.. O valor inicial é uma cadeia nula.

### ID da instância

Identificador de instância. Um valor binário (MQBYTE24) de comprimento MQ\_OBJECT\_INSTANCE\_ID\_LENGTH. O valor inicial é MQOII\_NONE.

### comprimento lógico

Comprimento lógico ou pretendido de dados da mensagem que seguem este cabeçalho. O valor inicial é zero.

### deslocamento lógico

Deslocamento lógico para os dados da mensagem que seguem, a ser interpretado no contexto dos dados como um todo, no destino final. O valor inicial é zero.

### deslocamento lógico 2

Extensão de alta ordem para a compensação lógica. O valor inicial é zero.

### tipo de referência

Tipo de referência.. O valor inicial é uma cadeia nula.

### ambiente de origem

Ambiente da origem. O valor inicial é uma cadeia nula.



## nome de origem

Nome da origem de dados. O valor inicial é uma cadeia nula.

## Construtores

### **ImqReferenceHeader ();**

O construtor padrão..

### **ImqReferenceHeader (const ImqReferenceHeader & header );**

O construtor de cópia..

## Métodos ImqItem sobrecarregados

### **virtual ImqBoolean copyOut ( ImqMessage & msg );**

Insere uma estrutura de dados MQRMH no buffer de mensagem no início, movendo os dados da mensagem existentes mais adiante e configura o formato *msg* para MQFMT\_REF\_MSG\_HEADER

Consulte a descrição do método de classe ImqHeader em [“Classe C++ ImqHeader” na página 1893](#) , para obter detalhes adicionais

### **virtual ImqBoolean pasteIn ( ImqMessage & msg );**

Lê uma estrutura de dados MQRMH do buffer de mensagem.

Para ser bem-sucedido, o formato ImqMessage deve ser MQFMT\_REF\_MSG\_HEADER.

Consulte a descrição do método de classe ImqHeader em [“Classe C++ ImqHeader” na página 1893](#) , para obter detalhes adicionais

## Métodos de objeto (público)

### **void operator = (const ImqReferenceHeader & header );**

Copia os dados de instância do *header*, substituindo os dados de instância existentes

### **ImqString destinationEnvironment () const;**

Retorna uma cópia do ambiente de destino..

### **void setDestinationEnvironment (const char \* environment = 0);**

Configura o ambiente de destino

### **ImqString destinationName () const;**

Retorna uma cópia do nome de destino..

### **void setDestinationName (const char \* name = 0);**

Configura o nome do destino

### **ImqBinary instanceId () const;**

Retorna uma cópia do ID da instância

### **ImqBoolean setInstanceId (const ImqBinary & id );**

Configura o ID da instância O comprimento de dados do *token* deve ser 0 ou MQ\_OBJECT\_INSTANCE\_ID\_LENGTH. Esse método retornará TRUE se for bem-sucedido

### **void setInstanceId (const MQBYTE24 id = 0);**

Configura o ID da instância *id* pode ser zero, que é o mesmo que especificar MQOII\_NONE. Se *id* for diferente de zero, ele deverá abordar MQ\_OBJECT\_INSTANCE\_ID\_LENGTH bytes de dados binários. Ao usar valores predefinidos, como MQOII\_NONE, pode ser necessário fazer uma conversão para assegurar uma correspondência de assinatura, por exemplo (MQBYTE \*) MQOII\_NONE

### **MQLONG logicalLength () const;**

Retorna o comprimento lógico..

### **void setLogicalLength (const MQLONG length );**

Configura o comprimento lógico

### **MQLONG logicalOffset () const;**

Retorna o deslocamento lógico

**void setLogicalOffset (const MQLONG *offset* );**

Configura o deslocamento lógico

**MQLONG logicalOffset2 () const;**

Retorna o deslocamento lógico 2.

**void setLogicalOffset2 (const MQLONG *offset* );**

Configura o deslocamento lógico 2.

**ImqString referenceType () const;**

Retorna uma cópia do tipo de referência..

**void setReferenceType (const char \* *name* = 0);**

Configura o tipo de referência

**ImqString sourceEnvironment () const;**

Retorna uma cópia do ambiente de origem..

**void setSourceEnvironment (const char \* *environment* = 0);**

Configura o ambiente de origem

**ImqString sourceName () const;**

Retorna uma cópia do nome de origem.

**void setSourceName (const char \* *name* = 0);**

Configura o nome de origem

## Dados do objeto (protegidos).

**MQRMH *omqrmh***

A estrutura de dados do MQRMH

## Códigos de Razão

- MQRC\_BINARY\_DATA\_LENGTH\_ERROR
- MQRC\_STRUC\_LENGTH\_ERROR
- MQRC\_STRUC\_ID\_ERROR
- MQRC\_INSUFENT\_DATA
- MQRC\_INCONSISTENT\_FORMAT
- MQRC\_ENCODING\_ERROR

## Classe C++ ImqString

Esta classe fornece armazenamento e manipulação de sequência de caracteres para sequências terminadas por nulo..

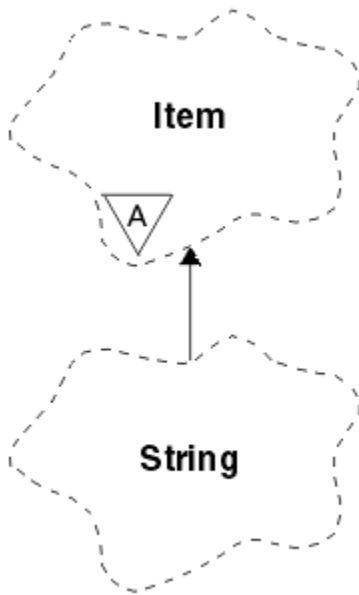


Figura 35. Classe ImqString

Utilize um ImqString no lugar de um **char \*** na maioria das situações em que um parâmetro chama um **char \***.

- [“Atributos do Objeto”](#) na página 1951
- [“Construtores”](#) na página 1951
- [“Métodos de classe \(público\).”](#) na página 1952
- [“Métodos ImqItem sobrecarregados”](#) na página 1952
- [“Métodos de objeto \(público\)”](#) na página 1952
- [“Métodos de objeto \(protegidos\)”](#) na página 1955
- [“Códigos de Razão”](#) na página 1955

## Atributos do Objeto

### caracteres

Caracteres no **armazenamento** que precedem um nulo final.

### comprimento

Número de bytes nos **caracteres**. Se não houver **armazenamento**, o **comprimento** será zero. O valor inicial é zero.

### Armazenamento

Uma matriz volátil de bytes de tamanho arbitrário. Um nulo final deve estar sempre presente no **armazenamento** após os **caracteres**, para que o final dos **caracteres** possa ser detectado. Os métodos asseguram que essa situação seja mantida, mas asseguram, ao configurar bytes na matriz diretamente, que um nulo final exista após modificação. Inicialmente, não há atributo **storage**.

## Construtores

### ImqString( );

O construtor padrão..

### ImqString(const ImqString & string );

O construtor de cópia..

### ImqString(const char c );

Os **caracteres** compreendem c.

**ImqString(const char \* text );**

Os **caracteres** são copiados de *text*.

**ImqString(const void \* buffer, const size\_t length );**

Copia *comprimento* bytes a partir de *buffer* e os designa aos **caracteres**. A substituição é feita para quaisquer caracteres nulos copiados. O caractere de substituição é um ponto (.). Nenhuma consideração especial é dada a quaisquer outros caracteres não imprimíveis ou não-exibíveis copiados.

## Métodos de classe (público).

**cópia ImqBoolean estática (char \* destination-buffer, const size\_t length, const char \* source-buffer, const char pad = 0);**

Copia até *length* bytes de *source-buffer* para *destination -buffer*. Se o número de caracteres em *source-buffer* for insuficiente, preencha o espaço restante em *destination-buffer* com caracteres *pad*. *source-buffer* pode ser zero. *destination-buffer* pode ser zero se *length* também for zero. Todos os códigos de erro são perdidos. Esse método retornará TRUE se for bem-sucedido.

**cópia ImqBoolean estática (char \* destination-buffer, const size\_t length, const char \* source-buffer, ImqError & error-object, const char pad = 0);**

Copia até *length* bytes de *source-buffer* para *destination -buffer*. Se o número de caracteres em *source-buffer* for insuficiente, preencha o espaço restante em *destination-buffer* com caracteres *pad*. *source-buffer* pode ser zero. *destination-buffer* pode ser zero se *length* também for zero. Todos os códigos de erro são configurados em *error-object*. Esse método retornará TRUE se for bem-sucedido.

## Métodos ImqItem sobrecarregados

**virtual ImqBoolean copyOut ( ImqMessage & msg );**

Copia os **caracteres** no buffer de mensagem, substituindo qualquer conteúdo existente. Configura o *msg format* para MQFMT\_STRING..

Consulte a descrição do método da classe-pai para obter detalhes adicionais.

**virtual ImqBoolean pasteIn ( ImqMessage & msg );**

Configura os **caracteres** transferindo os dados restantes do buffer de mensagem, substituindo os **caracteres** existentes.

Para ser bem sucedido, a **codificação** do objeto *msg* deve ser MQENC\_NATIVE. Recuperar mensagens com MQGMO\_CONVERT para MQENC\_NATIVE.

Para ser bem-sucedido, o ImqMessage **format** deve ser MQFMT\_STRING..

Consulte a descrição do método da classe-pai para obter detalhes adicionais.

## Métodos de objeto (público)

**char & operator [] (const size\_t offset ) const;**

Faz referência ao caractere no deslocamento *offset* no **armazenamento**.. Assegure-se de que o byte relevante exista e seja endereçável.

**Operador ImqString () (const size\_t offset, const size\_t length = 1) const;**

Retorna uma subsequência copiando bytes dos **caracteres** começando em *offset*. Se *length* for zero, retornará o restante dos **caracteres**. Se a combinação de *offset* e *length* não produzir uma referência dentro dos **caracteres**, retornará um ImqStringvazio.

**operador void = (const ImqString & string );**

Copia dados de instância de *sequência*, substituindo os dados de instância existentes..

**Operador ImqString + (const char c ) const;**

Retorna o resultado de anexar *c* aos **caracteres**.

**Operador ImqString + (const char \* text) const;**

Retorna o resultado de anexar *text* aos **caracteres**. Isso também pode ser invertido Por exemplo:

```
strOne + "string two" ;  
"string one" + strTwo ;
```

**Nota:** Embora a maioria dos compiladores aceite **strOne + "string two"**; Microsoft O Visual C++ requer **strOne + (char \*) "string two"**;

**Operador ImqString + (const ImqString & string1) const;**

Retorna o resultado de anexar *string1* aos **caracteres**.

**Operador ImqString + (const double number) const;**

Retorna o resultado de anexar *number* aos **caracteres** após a conversão em texto.

**Operador ImqString + (const long number) const;**

Retorna o resultado de anexar *number* aos **caracteres** após a conversão em texto.

**operador void + = (const char c);**

Anexa *c* aos **caracteres**..

**operador void + = (const char \* texto);**

Anexa *text* aos **caracteres**.

**operador void + = (const ImqString & string);**

Anexa *sequência* aos **caracteres**.

**operador void + = (const double number);**

Anexa *number* aos **caracteres** após a conversão para texto.

**operador void + = (const long number);**

Anexa *number* aos **caracteres** após a conversão para texto.

**operador char \* () const;**

Retorna o endereço do primeiro byte no **armazenamento**.. Esse valor pode ser zero e é volátil. Use este método apenas para propósitos somente leitura

**Operador ImqBoolean < (const ImqString & string) const;**

Compara os **caracteres** com aqueles de *sequência* usando o método **compare** .. O resultado é TRUE se for menor que e FALSE se for maior ou igual a.

**Operador ImqBoolean > (const ImqString & string) const;**

Compara os **caracteres** com aqueles de *sequência* usando o método **compare** .. O resultado é TRUE se maior que e FALSE se menor ou igual a.

**ImqBoolean operador < = (const ImqString & cadeia) const;**

Compara os **caracteres** com aqueles de *sequência* usando o método **compare** .. O resultado é TRUE se menor ou igual a e FALSE se maior que.

**Operador ImqBoolean > = (const ImqString & string) const;**

Compara os **caracteres** com aqueles de *sequência* usando o método **compare** .. O resultado é TRUE se maior ou igual a e FALSE se menor que.

**Operador ImqBoolean == (const ImqString & string) const;**

Compara os **caracteres** com aqueles de *sequência* usando o método **compare** .. Retorna TRUE ou FALSE.

**Operador ImqBoolean != (const ImqString & string) const;**

Compara os **caracteres** com aqueles de *sequência* usando o método **compare** .. Retorna TRUE ou FALSE.

**comparação curta (const ImqString & string) const;**

Compara os **caracteres** com os de *sequência*. O resultado será zero se os **caracteres** forem iguais, negativo se menor que e positivo se maior que. Comparação faz distinção entre maiúsculas e minúsculas. Um ImqString nulo é considerado como menor que um ImqString não nulo.

**ImqBoolean copyOut(char \* buffer, const size\_t length, const char pad = 0);**

Copia até *comprimento* bytes dos **caracteres** para o *buffer*. Se o número de **caracteres** for insuficiente, preencha o espaço restante no *buffer* com caracteres *preenchimento*. *buffer* pode ser zero se *length* também for zero. Ele retornará TRUE se for bem-sucedido

**size\_t copyOut(& número ) const;**

Configura *number* dos **caracteres** após a conversão de texto e retorna o número de caracteres envolvidos na conversão. Se for zero, nenhuma conversão foi executada e *number* não está configurado. Uma sequência de caracteres conversíveis deve começar com os seguintes valores:

```
<blank(s)>
<+|->
digit(s)
```

**size\_t copyOut( ImqString & token, const char c = ") const;**

Se os **caracteres** contiverem um ou mais caracteres diferentes de *c*, identificará um token como a primeira sequência contígua desses caracteres. Nesse caso, o *token* é configurado para essa sequência e o valor retornado é a soma do número de caracteres iniciais *c* e o número de bytes na sequência. Caso contrário, retorna zero e não configura *token*.

**size\_t cutOut(long & number );**

Configura *number* como para o método **copy**, mas também remove de **caracteres** o número de bytes indicado pelo valor de retorno. Por exemplo, a sequência mostrada no exemplo a seguir pode ser cortada em três números usando **cutOut** (*number*) três vezes:

```
strNumbers = "-1 0 +55 "
while ( strNumbers.cutOut( number ) );
number becomes -1, then 0, then 55
leaving strNumbers == " "
```

**size\_t cutOut( ImqString & token, const char c = " " )**

Configura *token* como para o método **copyOut** e remove de **caracteres** os caracteres *strToken* e também quaisquer caracteres *c* que precedem os caracteres *token*. Se *c* não for um espaço em branco, removerá os caracteres *c* que obtiverem diretamente os caracteres *token*. Retorna o número de caracteres removidos.. Por exemplo, a sequência mostrada no exemplo a seguir pode ser cortada em três tokens usando **cutOut** (*token*). três vezes:

```
strText = " Program Version 1.1 "
while ( strText.cutOut( token ) );
// token becomes "Program", then "Version",
// then "1.1" leaving strText == " "
```

O exemplo a seguir mostra como analisar um nome de caminho do DOS:

```
strPath = "C:\OS2\BITMAP\OS2LOGO.BMP"
strPath.cutOut( strDrive, ':' );
strPath.stripLeading( ':' );
while ( strPath.cutOut( strFile, '\\' ) );
// strDrive becomes "C".
// strFile becomes "OS2", then "BITMAP",
// then "OS2LOGO.BMP" leaving strPath empty.
```

**ImqBoolean (const ImqString & string );**

Procura uma correspondência exata para *string* em qualquer lugar dentro dos **caracteres**. Se nenhuma correspondência for localizada, ela retornará FALSE Caso contrário, retorna TRUE. Se *string* for nulo, ele retornará TRUE.

**ImqBoolean find (const ImqString & string, size\_t & offset );**

Procura uma correspondência exata para *string* em algum lugar dentro dos **caracteres** do deslocamento *offset* em diante. Se *string* for nulo, ele retorna TRUE sem atualizar *offset*. Se nenhuma correspondência for localizada, ela retornará FALSE (o valor de *offset* pode ter sido aumentado) Se uma correspondência for localizada, ela retornará TRUE e atualiza *offset* para o deslocamento de *string* dentro dos **caracteres**.

**comprimento de size\_t () const;**

Retorna o **length**.

**ImqBoolean pasteIn(const double number, const char \* format = "%f");**

Anexa *number* aos **caracteres** após a conversão para texto. Ele retornará TRUE se for bem-sucedido

O *format* de especificação é usado para formatar a conversão de ponto flutuante Se especificado, ele deverá ser adequado para uso com **printf** e números de ponto flutuante, por exemplo **%3f**.

**ImqBoolean pasteIn(const long number );**

Anexa *number* aos **caracteres** após a conversão para texto. Ele retornará TRUE se for bem-sucedido

**ImqBoolean pasteIn(const void \* buffer, const size\_t length );**

Anexa *length* bytes de *buffer* aos **caracteres** e inclui um nulo final. Substitui quaisquer caracteres nulos copiados. O caractere de substituição é um ponto (.). Nenhuma consideração especial é dada a quaisquer outros caracteres não-imprimíveis ou não-exibíveis copiados. Esse método retornará TRUE se for bem-sucedido

**ImqBoolean configurado (const char \* buffer, const size\_t length );**

Configura os **caracteres** de um campo de caractere de comprimento fixo, que pode conter um nulo. Anexa um nulo aos caracteres do campo de comprimento fixo, se necessário. Esse método retornará TRUE se for bem-sucedido

**ImqBoolean setStorage(const size\_t length );**

Aloca (ou realoca) o **armazenamento**. Preserva quaisquer **caracteres** originais, incluindo qualquer nulo final, se ainda houver espaço para eles, mas não inicializa nenhum armazenamento adicional.

Esse método retornará TRUE se for bem-sucedido

**size\_t storage () const;**

Retorna o número de bytes no **armazenamento**

**size\_t stripLeading(const char c = " ");**

Remove os caracteres iniciais *c* dos **caracteres** e retorna o número removido.

**size\_t stripTrailing(const char c = " ");**

Remove os caracteres finais *c* dos **caracteres** e retorna o número removido.

**ImqString upperCase() const;**

Retorna uma cópia maiúscula dos **caracteres**.

**Métodos de objeto (protegidos)****ImqBoolean assign ( const ImqString & string );**

Equivalente ao método **operator =** equivalente, mas não virtual. Ele retornará TRUE se for bem-sucedido

**Códigos de Razão**

- MQRD\_DATA\_TRUNCATED
- MQRD\_NULL\_POINTER
- MQRD\_STORAGE\_NOT\_AVAILABLE
- MQRD\_BUFFER\_ERROR
- MQRD\_INCONSISTENT\_FORMAT

## Classe C++ ImqTrigger

Essa classe encapsula a estrutura de dados MQTM (mensagem do acionador).

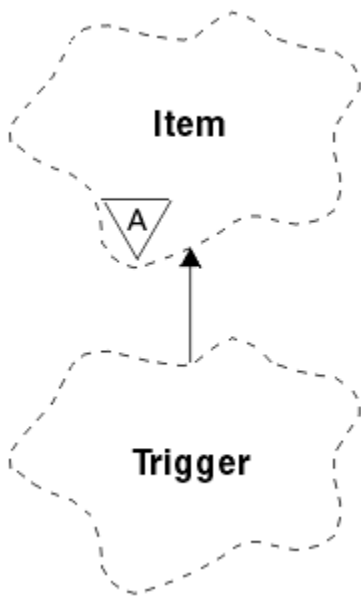


Figura 36. Classe ImqTrigger

Os objetos desta classe geralmente são utilizados por um programa monitor acionador. A tarefa de um programa do monitor acionador é aguardar essas mensagens específicas e agir sobre elas para assegurar que outros aplicativos IBM MQ sejam iniciados quando as mensagens estiverem esperando por elas

Consulte o programa de amostra IMQSTRG para um exemplo de uso.

- [“Atributos do Objeto” na página 1956](#)
- [“Construtores” na página 1957](#)
- [“Métodos ImqItem sobrecarregados” na página 1957](#)
- [“Métodos de objeto \(público\)” na página 1957](#)
- [“Dados do objeto \(protegidos\).” na página 1958](#)
- [“Códigos de Razão” na página 1958](#)

### Atributos do Objeto

#### ID do aplicativo

A identidade do aplicativo que enviou a mensagem O valor inicial é uma cadeia nula.

#### tipo de aplicativo

O tipo de aplicativo que enviou a mensagem O valor inicial é zero. Os seguintes valores adicionais são possíveis:

- MQAT\_AIX
- MQAT\_CICS
- MQAT\_DOS
- MQAT\_IMS
- MQAT\_MVS
- MQAT\_NOTES\_AGENT
- MQAT\_OS2
- MQAT\_OS390
- MQAT\_OS400



- MQAT\_UNIX
- MQAT\_WINDOWS
- MQAT\_WINDOWS\_NT
- MQAT\_USER\_FIRST
- MQAT\_USER\_LAST

### Dados do ambiente

Dados do ambiente para o processo O valor inicial é uma cadeia nula.

### nome do processo

Nome do processo. O valor inicial é uma cadeia nula.

### nome da fila

Nome da fila a ser iniciada.. O valor inicial é uma cadeia nula.

### dados do acionador

Dados do acionador para o processo O valor inicial é uma cadeia nula.

### dados do usuário

Dados do usuário para o processo.. O valor inicial é uma cadeia nula.

## Construtores

### ImqTrigger();

O construtor padrão..

### ImqTrigger(const ImqTrigger & acionador);

O construtor de cópia..

## Métodos ImqItem sobrecarregados

### virtual ImqBoolean copyOut ( ImqMessage & msg );

Grava uma estrutura de dados do MQTM no buffer de mensagem, substituindo qualquer conteúdo existente Configura o formato *msg* para MQFMT\_TRIGGER..

Consulte a descrição do método de classe ImqItem em [“Classe C++ ImqItem” na página 1897](#) , para obter detalhes adicionais

### virtual ImqBoolean pasteIn ( ImqMessage & msg );

Lê uma estrutura de dados MQTM do buffer de mensagem.

Para ser bem-sucedido, o formato de ImqMessage deve ser MQFMT\_TRIGGER

Consulte a descrição do método de classe ImqItem em [“Classe C++ ImqItem” na página 1897](#) , para obter detalhes adicionais

## Métodos de objeto (público)

### operador void = (acionador const ImqTrigger & );

Copia os dados de instância do *acionador*, substituindo os dados de instância existentes

### ImqString applicationId () const;

Retorna uma cópia do ID do aplicativo..

### void setApplicationId (const char \* id );

Configura o ID do aplicativo

### MQLONG applicationType () const;

Retorna o tipo de aplicativo..

### void setApplicationTipo (const MQLONG type );

Configura o tipo de aplicativo

### ImqBoolean copyOut ( MQTMC2 \* ptmc2 );

Encapsula a estrutura de dados do MQTM, que é aquela recebida nas filas de inicialização Preenche uma estrutura de dados MQTMC2 equivalente fornecida pelo responsável pela chamada e configura

o campo QMgrName (que não está presente na estrutura de dados do MQTM) para todos os espaços em branco. A estrutura de dados do MQTMC2 é tradicionalmente usada como um parâmetro para aplicativos iniciados por um monitor acionador. Esse método retornará TRUE se for bem-sucedido.

**ImqString environmentData () const;**

Retorna uma cópia dos dados do ambiente..

**void setEnvironmentData (const char \* data );**

Configura os dados do ambiente

**ImqString processName () const;**

Retorna uma cópia do nome do processo..

**void setProcessName (const char \* name );**

Configura o nome do processo, preenchido com espaços em branco para 48 caracteres.

**ImqString queueName (). const;**

Retorna uma cópia do nome da fila..

**void setQueueNome (const char \* name );**

Configura o nome da fila, preenchendo com espaços em branco para 48 caracteres..

**ImqString triggerData () const;**

Retorna uma cópia dos dados do acionador

**void setTriggerData (const char \* data );**

Configura os dados do acionador..

**ImqString userData () const;**

Retorna uma cópia dos dados do usuário..

**void setUserData (const char \* data );**

Configura os dados do usuário

## Dados do objeto (protegidos).

### MQTM *omqtm*

A estrutura de dados do MQTM

## Códigos de Razão

- MQRC\_NULL\_POINTER
- MQRC\_INCONSISTENT\_FORMAT
- MQRC\_ENCODING\_ERROR
- MQRC\_STRUC\_ID\_ERROR

## Classe C++ do cabeçalho ImqWork

Essa classe contém recursos específicos da estrutura de dados MQWIH.

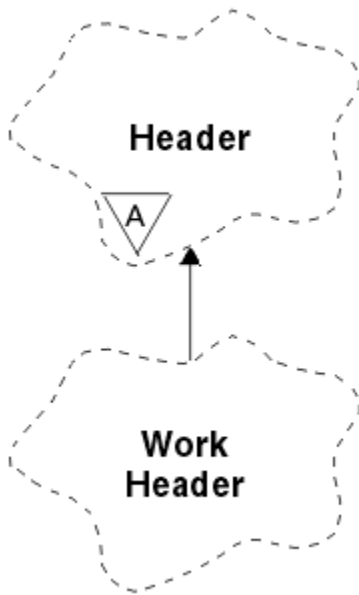


Figura 37. Classe de cabeçalho *ImqWork*

Os objetos dessa classe são usados por aplicativos que colocam mensagens na fila gerenciadas pelo z/OS Workload Manager

- [“Atributos do Objeto”](#) na página 1959
- [“Construtores”](#) na página 1959
- [“Métodos \*ImqItem\* sobrecarregados”](#) na página 1959
- [“Métodos de objeto \(público\)”](#) na página 1960
- [“Dados do objeto \(protegidos\).”](#) na página 1960
- [“Códigos de Razão”](#) na página 1960

## Atributos do Objeto

### **message token (token de mensagem)**

Token de mensagem para o z/OS Workload Manager de comprimento `MQ_MSG_TOKEN_LENGTH`. O valor inicial é `MQMTOK_NONE`.

### **Nome do serviço**

O nome de 32 caracteres de um processo O nome está inicialmente em branco..

### **etapa de serviço**

O nome de 8 caracteres de uma etapa dentro do processo O nome está inicialmente em branco..

## Construtores

### ***ImqWorkHeader* ();**

O construtor padrão..

### ***ImqWorkHeader* (const *ImqWorkHeader* & *header* );**

O construtor de cópia..

## Métodos *ImqItem* sobrecarregados

### **virtual *ImqBoolean* copyOut( *ImqMessage* & *msg* );**

Insere uma estrutura de dados `MQWIH` no início do buffer de mensagem, movendo os dados da mensagem existente mais adiante, e configura o *msg* **formato** para `MQFMT_WORK_INFO_HEADER`

Consulte a descrição do método da classe-pai para obter mais detalhes

### **virtual ImqBoolean pasteIn( ImqMessage & msg );**

Lê uma estrutura de dados MQWIH do buffer de mensagem.

Para ser bem sucedido, a codificação do objeto *msg* deve ser MQENC\_NATIVE Recuperar mensagens com MQGMO\_CONVERT para MQENC\_NATIVE.

O formato ImqMessage deve ser MQFMT\_WORK\_INFO\_HEADER (MQFMT\_WORK\_INFO\_HEADER)

Consulte a descrição do método da classe-pai para obter mais detalhes

## **Métodos de objeto (público)**

### **operador void = (const ImqWorkcabeçalho & cabeçalho );**

Copia os dados de instância do *header*, substituindo os dados de instância existentes

### **ImqBinary messageToken () const;**

Retorna o **token de mensagem**

### **ImqBoolean setMessageToken (const ImqBinary & );**

Configura o **token da mensagem** O comprimento de dados do *token* deve ser zero ou MQ\_MSG\_TOKEN\_LENGTH. Ele retornará TRUE se for bem-sucedido

### **void setMessageToken (const MQBYTE16 token = 0);**

Configura o **token da mensagem** *token* pode ser zero, que é o mesmo que especificar MQMTOK\_NONE. Se *token* for diferente de zero, ele deverá endereçar MQ\_MSG\_TOKEN\_LENGTH bytes de dados binários.

Ao usar valores predefinidos como MQMTOK\_NONE, pode ser necessário fazer uma conversão para assegurar uma correspondência de assinatura; por exemplo, (MQBYTE \*) MQMTOK\_NONE.

### **ImqString serviceName () const;**

Retorna o **service name**, incluindo espaços em branco finais.

### **void setServiceName (const char \* name );**

Configura o **nome do serviço**

### **ImqString serviceStep () const;**

Retorna a **etapa de serviço**, incluindo espaços em branco finais.

### **void setServiceStep (const char \* step );**

Configura a **etapa de serviço**

## **Dados do objeto (protegidos).**

### **MQWIH omqwih**

A estrutura de dados do MQWIH

## **Códigos de Razão**

- MQRC\_BINARY\_DATA\_LENGTH\_ERROR

## **Propriedades de objetos IBM MQ classes for JMS**

---

Todos os objetos em IBM MQ classes for JMS possuem propriedades Diferentes propriedades se aplicam a diferentes tipos de objetos.. Diferentes propriedades possuem diferentes valores permitidos e os valores de propriedades simbólicas diferem entre a ferramenta de administração e o código do programa

O IBM MQ classes for JMS fornece recursos para configurar e consultar as propriedades de objetos usando a ferramenta de administração do IBM MQ JMS , o IBM MQ Explorer ou em um aplicativo Muitas das propriedades são relevantes apenas para um subconjunto específico dos tipos de objeto

Para obter informações sobre como você usa a ferramenta de administração do IBM MQ JMS , consulte [Configurando JMS objetos usando a ferramenta de administração](#)

[Tabela 868 na página 1961](#) fornece uma breve descrição de cada propriedade e mostra para cada propriedade a quais tipos de objeto ela se aplica Os tipos de objeto são identificados usando palavras-

chave; consulte [Configurando JMS objetos usando a ferramenta de administração](#) para obter uma explicação desses objetos.

Os números referem-se a notas no final da tabela Consulte também [“Dependências entre as propriedades de objetos do IBM MQ classes for JMS”](#) na página 1964.

Uma propriedade consiste em um par nome-valor no formato:

```
PROPERTY_NAME(property_value)
```

Os tópicos nesta seção listam, para cada propriedade, o nome da propriedade e uma breve descrição, e mostra os valores de propriedade válidos usados na ferramenta de administração e o método set que é usado para configurar o valor da propriedade em um aplicativo. Os tópicos também mostram os valores de propriedade válidos para cada propriedade e o mapeamento entre os valores de propriedade simbólica usados na ferramenta e seus equivalentes programáveis...

Os nomes de propriedades não fazem distinção entre maiúsculas e minúsculas e são restritos ao conjunto de nomes reconhecidos mostrado nesses tópicos.

*Tabela 868. Nomes de propriedades e tipos de objetos aplicáveis*

Propriedade	Formato curto	Tipo de objeto							
		CF	QCF	TCF	Q	T	XACF	XAQCF	XATCF
<a href="#">“APPLICATIONNAME”</a> na página 1966	APPNAME	Y	Y	Y			Y	Y	Y
<a href="#">“ASYNCEXCEPTION”</a> na página 1966	AEX	Y	Y	Y			Y	Y	Y
<a href="#">“BALOPTIONS”</a> na página 1967	OPTIONS	Y	Y	Y			Y	Y	Y
<a href="#">“BALTYPE”</a> na página 1968	Tipo	Y	Y	Y			Y	Y	Y
<a href="#">“BALTIMEOUT”</a> na página 1969	TIMEOUT	Y	Y	Y			Y	Y	Y
<a href="#">“BROKERCCDURSUBQ”</a> na página 1969 <sup>1</sup>	CCDSUB					Y			
<a href="#">“BROKERCCSUBQ”</a> na página 1970 <sup>1</sup>	CCSUB	Y		Y			Y		Y
<a href="#">“BROKERCONQ”</a> na página 1970 <sup>1</sup>	BCON	Y		Y			Y		Y
<a href="#">“BROKERDURSUBQ”</a> na página 1971 <sup>1</sup>	BDSUB					Y			
<a href="#">“BROKERPUBQ”</a> na página 1971 <sup>1</sup>	BPUB	Y		Y		Y	Y		Y
<a href="#">“BROKERPUBQMGR”</a> na página 1971 <sup>1</sup>	BPQM					Y			
<a href="#">“BROKERQMGR”</a> na página 1972 <sup>1</sup>	BQM	Y		Y			Y		Y
<a href="#">“BROKERSUBQ”</a> na página 1972 <sup>1</sup>	BSUB	Y		Y			Y		Y
<a href="#">“BROKERVER”</a> na página 1973 <sup>1</sup>	BVER	Y <sup>2</sup>		Y <sup>2</sup>		Y	Y		Y
<a href="#">“CCDTURL”</a> na página 1973 <sup>3</sup>	CCDT	Y	Y	Y			Y	Y	Y
<a href="#">“CCSID”</a> na página 1974	CCS	Y	Y	Y	Y	Y	Y	Y	Y
<a href="#">“CHANNEL”</a> na página 1974 <sup>3</sup>	CHAN	Y	Y	Y			Y	Y	Y
<a href="#">“CLEANUP”</a> na página 1975 <sup>1</sup>	CL	Y		Y			Y		Y
<a href="#">“CLEANUPINT”</a> na página 1975 <sup>1</sup>	CLINT	Y		Y			Y		Y
<a href="#">“connectionNameList”</a> na página 1976	CNLIST	Y	Y	Y					

Tabela 868. Nomes de propriedades e tipos de objetos aplicáveis (continuação)

Propriedade	Formato curto	Tipo de objeto							
		CF	QCF	TCF	Q	T	XACF	XAQCF	XATCF
<a href="#">“CLIENTRECONNECTOPTIONS” na página 1976</a>	CROPT	Y	Y	Y					
<a href="#">“CLIENTRECONNECTTIMEOUT” na página 1977</a>	CRT	Y	Y	Y					
<a href="#">“CLIENTID” na página 1978</a>	CID	Y <sup>2</sup>	Y	Y <sup>2</sup>			Y	Y	Y
<a href="#">“CLONESUPP” na página 1978</a>	CLS	Y		Y			Y		Y
<a href="#">“COMPHDR” na página 1979</a>	HC	Y		Y			Y		Y
<a href="#">“COMPMSG” na página 1979</a>	MC	Y	Y	Y			Y	Y	Y
<a href="#">“CONNOPT” na página 1980</a>	CNOPT	Y	Y	Y			Y	Y	Y
<a href="#">“CONNTAG” na página 1981</a>	CNTAG	Y	Y	Y			Y	Y	Y
<a href="#">“DESCRIÇÃO” na página 1981</a>	DESC	Y <sup>2</sup>	Y	Y <sup>2</sup>	Y	Y	Y	Y	Y
<a href="#">“DIRECTAUTH” na página 1981</a>	DAUTH	Y <sup>2</sup>		Y <sup>2</sup>					
<a href="#">“ENCODING” na página 1982</a>	ENC				Y	Y			
<a href="#">“EXPIRY” na página 1983</a>	EXP				Y	Y			
<a href="#">“FAILIFQUIESCE” na página 1983</a>	FIQ	Y	Y	Y	Y	Y	Y	Y	Y
<a href="#">“HOSTNAME” na página 1984</a>	HOST	Y <sup>2</sup>	Y	Y <sup>2</sup>			Y	Y	Y
<a href="#">“LOCALADDRESS” na página 1985</a>	LA	Y <sup>2</sup>	Y	Y <sup>2</sup>			Y	Y	Y
<a href="#">“MAPNAMESTYLE” na página 1985</a>	MNST	Y	Y	Y			Y	Y	Y
<a href="#">“MAXBUFFSIZE” na página 1986</a>	MBSZ	Y <sup>2</sup>		Y <sup>2</sup>					
<a href="#">“MDREAD” na página 1986</a>	MDR				Y	Y			
<a href="#">“MDWRITE” na página 1987</a>	MDW				Y	Y			
<a href="#">“MDMSGCTX” na página 1987</a>	MDCTX				Y	Y			
<a href="#">“MSGBATCHSZ” na página 1988<sup>1</sup></a>	MBS	Y	Y	Y			Y	Y	Y
<a href="#">“MSGBODY” na página 1989</a>	MBODY				Y	Y			
<a href="#">“MSGRETENTION” na página 1989</a>	MRET	Y	Y				Y	Y	
<a href="#">“MSGSELECTION” na página 1990<sup>1</sup></a>	MSEL	Y		Y			Y		Y
<a href="#">“MULTICAST” na página 1990</a>	MCAST	Y <sup>2</sup>		Y <sup>2</sup>		Y			
<a href="#">“OPTIMISTICPUBLICATION” na página 1991<sup>1</sup></a>	OPTPUB	Y		Y					
<a href="#">“OUTCOMENOTIFICATION” na página 1991<sup>1</sup></a>	NOTIFY	Y		Y					
<a href="#">“PERSISTENCE” na página 1992</a>	PER				Y	Y			
<a href="#">“POLLINGINT” na página 1993<sup>1</sup></a>	PINT	Y	Y	Y			Y	Y	Y
<a href="#">“PORT” na página 1993</a>	PORT	Y <sup>2</sup>	Y	Y <sup>2</sup>			Y	Y	Y
<a href="#">“PRIORITY” na página 1994</a>	PRI				Y	Y			

Tabela 868. Nomes de propriedades e tipos de objetos aplicáveis (continuação)

Propriedade	Formato curto	Tipo de objeto							
		CF	QCF	TCF	Q	T	XACF	XAQCF	XATCF
<a href="#">"PROCESSDURATION" na página 1994 <sup>1</sup></a>	PROCDUR	Y		Y					
<a href="#">"PROVIDERVERSION" na página 1995</a>	PVER	Y	Y	Y			Y	Y	Y
<a href="#">"PROXYHOSTNAME" na página 1997</a>	PHOST	Y <sup>2</sup>		Y <sup>2</sup>					
<a href="#">"PROXYPORT" na página 1997</a>	PPORT	Y <sup>2</sup>		Y <sup>2</sup>					
<a href="#">"PUBACKINT" na página 1998 <sup>1</sup></a>	PAI	Y		Y			Y		Y
<a href="#">"PUTASYNCALLOWED" na página 1998</a>	PAALD				Y	Y			
<a href="#">"QMANAGER" na página 1999</a>	QMGR	Y	Y	Y	Y		Y	Y	Y
<a href="#">"FILA" na página 1999</a>	QU				Y				
<a href="#">"READAHEADALLOWED" na página 2000</a>	RAALD				Y	Y			
<a href="#">"READAHEADCLOSEPOLICY" na página 2000</a>	RACP				Y	Y			
<a href="#">"RECEIVECCSID" na página 2001</a>	RCCS				Y	Y			
<a href="#">"RECEIVECONVERSION" na página 2001</a>	RCNV				Y	Y			
<a href="#">"RECEIVEISOLATION" na página 2002 <sup>1</sup></a>	RCVISOL	Y		Y					
<a href="#">"RECEXIT" na página 2002</a>	RCX	Y	Y	Y			Y	Y	Y
<a href="#">"RECEXITINIT" na página 2003</a>	RCXI	Y	Y	Y			Y	Y	Y
<a href="#">"REPLYTOSTYLE" na página 2003</a>	RTOST				Y	Y			
<a href="#">"RESCANINT" na página 2004 <sup>1</sup></a>	RINT	Y	Y				Y	Y	
<a href="#">"SECEXIT" na página 2004</a>	SCX	Y	Y	Y			Y	Y	Y
<a href="#">"SECEXITINIT" na página 2005</a>	SCXI	Y	Y	Y			Y	Y	Y
<a href="#">"SENDCHECKCOUNT" na página 2005</a>	SCC	Y	Y	Y			Y	Y	Y
<a href="#">"SENDEXIT" na página 2006</a>	SDX	Y	Y	Y			Y	Y	Y
<a href="#">"SENDEXITINIT" na página 2006</a>	SDXI	Y	Y	Y			Y	Y	Y
<a href="#">"SHARECONVALLOWED" na página 2007</a>	SCALD	Y	Y	Y			Y	Y	Y
<a href="#">"SPARSESUBS" na página 2008 <sup>1</sup></a>	SSUBS	Y		Y					
<a href="#">"SSLCIPHERSUITE" na página 2008</a>	SCPHS	Y	Y	Y			Y	Y	Y
<a href="#">"SSLCRL" na página 2009</a>	SCRL	Y	Y	Y			Y	Y	Y
<a href="#">"SSLFIPSREQUIRED" na página 2009</a>	SFIPS	Y	Y	Y			Y	Y	Y
<a href="#">"SSLPEERNAME" na página 2010</a>	SPEER	Y	Y	Y			Y	Y	Y
<a href="#">"SSLRESETCOUNT" na página 2010</a>	SRC	Y	Y	Y			Y	Y	Y

Tabela 868. Nomes de propriedades e tipos de objetos aplicáveis (continuação)

Propriedade	Formato curto	Tipo de objeto							
		CF	QCF	TCF	Q	T	XACF	XAQCF	XATCF
<a href="#">“STATREFRESHINT” na página 2010 <sup>1</sup></a>	SRI	Y		Y			Y		Y
<a href="#">“SUBSTORE” na página 2011 <sup>1</sup></a>	SS	Y		Y			Y		Y
<a href="#">“SYNCPOINTALLGETS” na página 2011</a>	SPAG	Y	Y	Y			Y	Y	Y
<a href="#">“TARGCLIENT” na página 2012</a>	TC				Y	Y			
<a href="#">“TARGCLIENTMATCHING” na página 2012</a>	TCM	Y	Y				Y	Y	
<a href="#">“TEMPMODEL” na página 2013</a>	TM	Y	Y				Y	Y	
<a href="#">“TEMPQPREFIX” na página 2013</a>	TQP	Y	Y				Y	Y	
<a href="#">“TEMPTOPICPREFIX” na página 2014</a>	TTP	Y		Y			Y		Y
<a href="#">“tçpico” na página 2014</a>	TOP					Y			
<a href="#">“TRANSPORT” na página 2015</a>	TRAN	Y <sup>2</sup>	Y	Y <sup>2</sup>			Y	Y	Y
<a href="#">“WILDCARDFORMAT” na página 2015</a>	WCFMT	Y		Y			Y		Y

**Nota:**

- Essa propriedade pode ser usada com a versão 70 do IBM MQ classes for JMS , mas não tem efeito para um aplicativo conectado a um gerenciador de fila do IBM WebSphere MQ 7.0 , a menos que a propriedade PROVIDERVERSION do connection factory seja configurada para um número de versão menor que 7
- Apenas as propriedades BROKERVER, CLIENTID, DESCRIPTION, DIRECTAUTH, hostname, localaddress, maxbuffsize, multicast, port, proxyhostname, PROXYPORT e TRANSPORT são suportadas para um objeto Factory ConnectionFactory ou TopicConnectionao usar uma conexão em tempo real com um broker.
- As propriedades CCDURL e CHANNEL de um objeto não devem ser configuradas ao mesmo tempo.

## Dependências entre as propriedades de objetos do IBM MQ classes for JMS

A validade de algumas propriedades é dependente dos valores específicos de outras propriedades

Essa dependência pode ocorrer nos seguintes grupos de propriedades:

- Propriedades do cliente
- Propriedades para uma conexão em tempo real com um broker
- Sequências de inicialização de saída..

### Propriedades do cliente

Para uma conexão com um gerenciador de filas, as propriedades a seguir são relevantes apenas se TRANSPORT tiver o valor CLIENT:

- HOSTNAME
- PORT
- CHANNEL
- LOCALADDRESS
- CCDURL
- CCSID



- COMPHDR
- COMPMSG
- REEXIT
- REEXITINIT
- SEEXIT
- SEEXITINIT
- SENDEXIT
- SENDEXITINIT
- SHARECONVALLOWED
- SSLCIPHERSUITE
- SSLCRL
- SSLFIPSREQUIRED
- SSLPEERNAME
- SSLRESETCOUNT
- APPLICATIONNAME

Não é possível configurar valores para essas propriedades usando a ferramenta de administração se TRANSPORT tiver o valor BIND.

TRANSPORT tem o valor CLIENT, o valor padrão da propriedade BROKERVER é V1 e o valor padrão da propriedade PORT é 1414. Se você configurar o valor de BROKERVER ou PORT explicitamente, uma mudança posterior no valor de TRANSPORT não substituirá suas opções.

#### **Propriedades para uma conexão em tempo real com um broker**

Apenas as seguintes propriedades serão relevantes se TRANSPORT tiver o valor DIRECT ou DIRECTHTTP:

- BROKERVER
- CLIENTID
- DESCRIÇÃO
- DIRECTAUTH
- HOSTNAME
- LOCALADDRESS
- MAXBUFFSIZE
- MULTICAST (suportado apenas para DIRECT)
- PORT
- PROXYHOSTNAME (suportado apenas para DIRECT)
- PROXYPORT (suportado apenas para DIRECT)

Se TRANSPORT tiver o valor DIRECT ou DIRECTHTTP, o valor padrão da propriedade BROKERVER será V2e o valor padrão da propriedade PORT será 1506. Se você configurar o valor de BROKERVER ou PORT explicitamente, uma mudança posterior no valor de TRANSPORT não substituirá suas opções.

#### **Sequências de inicialização de saída..**

Não configure qualquer uma das cadeias de inicialização de saída sem fornecer o nome de saída correspondente As propriedades de inicialização de saída são:

- REEXITINIT
- SEEXITINIT
- SENDEXITINIT

Por exemplo, especificar REEXITINIT(myString) sem especificar REEXIT(some.exit.classname) causa um erro.

### Referências relacionadas

[“TRANSPORT” na página 2015](#)

A natureza de uma conexão com um gerenciador de filas ou um broker

## APPLICATIONNAME

Um aplicativo pode configurar um nome que identifica sua conexão com o gerenciador de filas. Esse nome do aplicativo é mostrado no comando **DISPLAY CONN MQSC/PCF** (em que o campo é chamado **APPLTAG**) ou na exibição IBM MQ Explorer **Conexões de Aplicativos** (em que o campo é chamado **App name**).

### Objetos Aplicáveis

ConnectionFactory, QueueConnectionFactory, TopicConnectionFactory, XAConnectionFactory, XAQueueConnectionFactory, XATopicConnectionFactory.

Nome longo da ferramenta de administração do JMS : APPLICATIONNAME

Nome abreviado da ferramenta de administração JMS : APPNAME

### Acesso programático


Setters / getters

- MQConnectionFactory.setAppName ()
- MQConnectionFactory.getAppName ()

### Valores

Qualquer sequência válida que não tenha mais do que 28 caracteres. Nomes mais longos são ajustados para caber removendo nomes de pacotes principais, se necessário. Por exemplo, se a classe de chamada for com.example.MainApp, o nome completo será usado, mas se a classe de chamada for com.example.dictionaryAndThesaurus.multilingual.mainApp, o nome multilingual.mainApp será usado, pois ele é a combinação mais longa do nome da classe e o nome do pacote mais à direita que cabe no comprimento disponível.

Se o próprio nome da classe tiver mais de 28 caracteres de comprimento, ele será truncado para ajuste. Por exemplo, com.example.mainApplicationForSecondTestCase se torna mainApplicationForSecondTest.

 No z/OS, o APPNAME em:

- O modo de ligações é ignorado se configurado e, se configurado, só pode ser configurado como espaços em branco
- O modo de cliente pode ser configurado e usado

## ASYNCEXCEPTION

Essa propriedade determina se o IBM MQ classes for JMS informa um ExceptionListener somente quando uma conexão é interrompida ou quando qualquer exceção ocorre de forma assíncrona para uma chamada API JMS . Isso se aplica a todas as Conexões criadas a partir desse ConnectionFactory que possuem um ExceptionListener registrado..

### Objetos aplicáveis

ConnectionFactory, QueueConnectionFactory, TopicConnectionFactory, XAConnectionFactory, XAQueueConnectionFactory, XATopicConnectionFactory.

Nome longo da ferramenta de administração do JMS : ASYNCEXCEPTION

Nome abreviado da ferramenta de administração do JMS : AEX

## Acesso programático

Configuradores / Getters

- MQConnectionFactory.setAsyncExceções ()
- MQConnectionFactory.getAsyncExceções ()

## Valores

### ASYNCEXCEPTIONS\_ALL

Qualquer exceção detectada de forma assíncrona, fora do escopo de uma chamada de API síncrona, e todas as exceções de conexão interrompida são enviadas para o ExceptionListener

*Tabela 869. Todas as exceções assíncronas: ambientes e nomes de constantes relacionados*

Meio ambiente	Value
JMS Ferramenta de Administração	ALL
Programático	WMQCONSTANTS.ASYNCEXCEPTIONS_ALL = -1
IBM MQ Explorer	Todos(as)

### ASYNCEXCEPTIONS\_CONNECTIONBROKEN

Apenas exceções indicando uma conexão interrompida são enviadas para o ExceptionListener. Quaisquer outras exceções que ocorram durante o processamento assíncrono não são relatadas para o ExceptionListener, portanto, o aplicativo não é informado sobre essas exceções Este é o valor padrão do IBM MQ 8.0.0 Fix Pack 2. Consulte [JMS: mudanças do listener de exceção no IBM MQ 8.0](#).

*Tabela 870. Exceções indicando uma conexão interrompida: ambientes e nomes de constantes relacionados*

Meio ambiente	Value
JMS Ferramenta de Administração	CONNECTIONBROKEN
Programático	WMQCONSTANTS.ASYNCEXCEPTIONS_CONNECTIONBROKEN = 1
IBM MQ Explorer	Conexão interrompida

A constante adicional a seguir é definida:

- Em IBM MQ 8.0.0 Fix Pack 2: WMQCONSTANTS.ASYNCEXCEPTIONS\_DEFAULT = ASYNCEXCEPTIONS\_CONNECTIONBROKEN
- Antes de IBM MQ 8.0.0 Fix Pack 2: WMQCONSTANTS.ASYNCEXCEPTIONS\_DEFAULT = ASYNCEXCEPTIONS\_ALL

### Conceitos relacionados

[Exceções no IBM MQ classes for JMS](#)

## V 9.3.4 BALOPTIONS

Controla como os aplicativos do IBM MQ classes for JMS que usam o transporte do cliente são rebalanceadas em clusters uniformes

## Objetos Aplicáveis

ConnectionFactory, QueueConnectionFactory, TopicConnectionFactory, XAConnectionFactory, XAQueueConnectionFactory, XATopicConnectionFactory.

JMS nome longo da ferramenta de administração: **BALOPTIONS**

JMS nome abreviado da ferramenta de administração: **OPTIONS**

## Acesso programático

Setters / getters

- `MQConnectionFactory.setBalancingOptions()`
- `MQConnectionFactory.getBalancingOptions()`

## Valores

### IGNNONE

A manipulação normal de transações se aplica e os aplicativos não são solicitados a mover durante uma transação.

Esse valor é mapeado para IBM MQ *BalancingOption* `MQBNO_OPTIONS_NONE`.

### IGNTRANS

Os aplicativos podem ser solicitados para mover durante uma transação.

Esse valor é mapeado para IBM MQ *BalancingOption* `MQBNO_OPTIONS_IGNORE_TRANS`.

## V 9.3.4 BALTYPE

Controla como os aplicativos IBM MQ classes for JMS que usam o transporte do cliente podem ser rebalanceadas em um cluster uniforme

## Objetos Aplicáveis

ConnectionFactory, QueueConnectionFactory, TopicConnectionFactory, XAConnectionFactory, XAQueueConnectionFactory, XATopicConnectionFactory.

JMS nome longo da ferramenta de administração: **BALTYPE**

JMS nome abreviado da ferramenta de administração: **TYPE**

## Acesso programático

Setters / getters

- `MQConnectionFactory.setBalancingApplicationType()`
- `MQConnectionFactory.getBalancingApplicationType()`

## Valores

### SIMPLE

Aplica-se a manipulação padrão de aplicativos em um cluster uniforme

Esse valor é mapeado para IBM MQ *BalancingOption* `MQBNO_BALTYPE_SIMPLE`.

### REQUEST\_REPLY

O aplicativo não será solicitado para reconectar se um **MQPUT** não tiver sido balanceado por um **MQGET**, a menos que o período de tempo limite tenha decorrido.

Esse valor é mapeado para o IBM MQ *BalancingOption* `MQBNO_BALTYPE_REQREP` . .

## V 9.3.4 BALTIMEOUT

Controla como aplicativos IBM MQ classes for JMS que usam o transporte do cliente são rebalanceados em um cluster uniforme.

### Objetos Aplicáveis

ConnectionFactory, QueueConnectionFactory, TopicConnectionFactory, XAConnectionFactory, XAQueueConnectionFactory, XATopicConnectionFactory.

JMS nome longo da ferramenta de administração: **BALTIMEOUT**

JMS nome abreviado da ferramenta de administração: **TIMEOUT**

### Acesso programático

Setters / getters

- `MQConnectionFactory.setBalancingTimeout()`
- `MQConnectionFactory.getBalancingTimeout()`

### Valores

#### never

O aplicativo nunca atinge o tempo limite para os propósitos de rebalanceamento em um cluster uniforme

Esse valor é mapeado para IBM MQ *BalancingOption* MQBNO\_TIMEOUT\_NEVER.

#### IMEDIATO

O aplicativo atinge o tempo limite imediatamente para os propósitos de rebalanceamento em um cluster uniforme

Esse valor é mapeado para IBM MQ *BalancingOption* MQBNO\_TIMEOUT\_IMMEDIATE.

#### DEFAULT

O aplicativo atinge o tempo limite para os propósitos de rebalanceamento em um cluster uniforme após o período padrão de 10 segundos

Esse valor é mapeado para o IBM MQ *BalancingOption* MQBNO\_TIMEOUT\_AS\_DEFAULT.

#### nn

O aplicativo atinge o tempo limite com o propósito de rebalanceamento em um cluster uniforme após um período de *nn* segundos.

*nn* pode estar entre 1 e 9999999999.

## BROKERCCDURSUBQ

O nome da fila da qual as mensagens de assinatura durável são recuperadas para um ConnectionConsumer.

### Objetos aplicáveis

Tópico

Nome longo da ferramenta de administração do JMS : BROKERCCDURSUBQ

Nome abreviado da ferramenta de administração do JMS : CCDSUB

### Acesso programático

Setters / getters

- `MQTopic.setBrokerCCDurSubQueue()`

- MQTopic.getBrokerCCDurSubQueue()

## Valores

### **SYSTEM.JMS.D.CC.SUBSCRIBER.QUEUE**

Esse é o valor-padrão.

**Qualquer sequência válida**

## **BROKERCCSUBQ**

O nome da fila da qual as mensagens de assinatura não duráveis são recuperadas para um ConnectionConsumer.

### **Objetos aplicáveis**

ConnectionFactory, TopicConnectionFactory, XAConnectionFactory, XATopicConnectionFactory

Nome longo da ferramenta de administração do JMS : BROKERCCSUBQ

Nome abreviado da ferramenta de administração do JMS : CCSUB

### **Acesso programático**

Setters / getters

- MQConnectionFactory.setBrokerCCSubQueue()
- MQConnectionFactory.getBrokerCCSubQueue()

## Valores

### **SYSTEM.JMS.ND.CC.SUBSCRIBER.QUEUE**

Esse é o valor-padrão.

**Qualquer sequência válida**

## **BROKERCONQ**

O nome da fila de controle do broker..

### **Objetos Aplicáveis**

ConnectionFactory, TopicConnectionFactory, XAConnectionFactory, XATopicConnectionFactory

Nome longo da ferramenta de administração do JMS : BROKERCONQ

Nome abreviado da ferramenta de administração do JMS : BCON

### **Acesso programático**

Setters / getters

- MQConnectionFactory.setBrokerControlQueue()
- MQConnectionFactory.getBrokerControlQueue()

## Valores

### **SYSTEM.BROKER.CONTROL.QUEUE**

Esse é o valor-padrão.

**Qualquer sequência válida**

## BROKERDURSUBQ

Quando o IBM MQ classes for JMS está sendo usado no modo de migração do provedor de sistemas de mensagens do IBM MQ , essa propriedade especifica o nome da fila a partir da qual as mensagens de assinaturas duráveis são recuperadas.

### Objetos aplicáveis

Tópico

Nome longo da ferramenta de administração do JMS : BROKERDURSUBQ

Nome abreviado da ferramenta de administração do JMS : BDSUB

### Acesso programático

Setters / getters

- MQTopic.setBrokerDurSubQueue()
- MQTopic.getBrokerDurSubQueue()

### Valores

#### **SYSTEM.JMS.D.SUBSCRIBER.QUEUE**

Esse é o valor-padrão.

#### **Qualquer sequência válida**

Começando com SYSTEM.JMS.D

#### **Tarefas relacionadas**

Configurando a propriedade JMS **PROVIDERVERSION**

## BROKERPUBQ

O nome da fila para onde as mensagens publicadas são enviadas (a fila de fluxo).

### Objetos Aplicáveis

ConnectionFactory, TopicConnectionFactory, Tópico, XAConnectionFactory, XATopicConnectionFactory

Nome longo da ferramenta de administração do JMS : BROKERPUBQ

Nome abreviado da ferramenta de administração do JMS : BPUB

### Acesso programático

Setters / getters

- MQConnectionFactory.setBrokerPubQueue
- MQConnectionFactory.getBrokerPubQueue

### Valores

#### **SYSTEM.BROKER.DEFAULT.STREAM**

Esse é o valor-padrão.

#### **Qualquer sequência válida**

## BROKERPUBQMGR

O nome do gerenciador de filas que possui a fila em que mensagens publicadas sobre o tópico são enviadas.

## Objetos Aplicáveis

Tópico

Nome longo da ferramenta de administração do JMS : BROKERPUBQMGR

Nome abreviado da ferramenta de administração do JMS : BPQM

## Acesso programático

Setters / getters

- MQTopic.setBrokerPubQueueManager()
- MQTopic.getBrokerPubQueueManager()

## Valores

nulo

Esse é o valor-padrão.

Qualquer sequência válida

## BROKERQMGR

O nome do gerenciador de filas no qual o broker está em execução.

## Objetos Aplicáveis

ConnectionFactory, TopicConnectionFactory, XAConnectionFactory, XATopicConnectionFactory

Nome longo da ferramenta de administração do JMS : BROKERQMGR

Nome abreviado da ferramenta de administração do JMS : BQM.

## Acesso programático

Setters / getters

- MQConnectionFactory.setBrokerQueueManager()
- MQConnectionFactory.getBrokerQueueManager()

## Valores

nulo

Esse é o valor-padrão.

Qualquer sequência válida

## BROKERSUBQ

Quando o IBM MQ classes for JMS está sendo usado no modo de migração do provedor de sistemas de mensagens do IBM MQ , essa propriedade especifica o nome da fila a partir da qual as mensagens de assinaturas não duráveis são recuperadas.

## Objetos Aplicáveis

ConnectionFactory, TopicConnectionFactory, XAConnectionFactory, XATopicConnectionFactory

Nome longo da ferramenta de administração do JMS : BROKERSUBQ

Nome abreviado da ferramenta de administração do JMS : BSUB



## Acesso programático

Setters / getters

- MQConnectionFactory.setBrokerSubQueue()
- MQConnectionFactory.getBrokerSubQueue()

## Valores

### SYSTEM.JMS.ND.SUBSCRIBER.QUEUE

Esse é o valor-padrão.

### Qualquer sequência válida

Começando com SYSTEM.JMS.ND

### Tarefas relacionadas

Configurando a propriedade JMS **PROVIDERVERSION**

## BROKERVER

A versão do broker que está sendo usada.

### Objetos Aplicáveis

ConnectionFactory, TopicConnectionFactory, Tópico, XAConnectionFactory, XATopicConnectionFactory

Nome longo da ferramenta de administração do JMS : BROKERVER

Nome abreviado da ferramenta de administração do JMS : BVER

## Acesso programático

Setters / getters

- MQConnectionFactory.setBrokerVersão ()
- MQConnectionFactory.getBrokerVersão ()

## Valores

### V1

Para usar um broker de publicação / assinatura do IBM MQ ou para usar um broker de IBM MQ Integrator, WebSphere Event Broker, WebSphere Business Integration Event Broker ou WebSphere Business Integration Message Broker no modo de compatibilidade. Esse será o valor padrão se TRANSPORT estiver configurado como BIND ou CLIENT.

### V2

Para usar um broker do IBM MQ Integrator WebSphere Event Broker, WebSphere Business Integration Event Broker ou WebSphere Business Integration Message Broker no modo nativo.. Este é o valor padrão se TRANSPORT estiver configurado como DIRECT ou DIRECTHTTP...

### não especificado

Após o broker ter migrado de V6 para V7, configure essa propriedade para que os cabeçalhos RFH2 não sejam mais usados. Após a migração, essa propriedade não é mais relevante.

## CCDTURL

Um Localizador Uniforme de Recursos (URL) que identifica o nome e o local do arquivo que contém a tabela de definição de canal de cliente e especifica como o arquivo pode ser acessado.

## Objetos Aplicáveis

ConnectionFactory, QueueConnectionFactory, TopicConnectionFactory, XAConnectionFactory, XAQueueConnectionFactory, XATopicConnectionFactory.

Nome longo da ferramenta de administração do JMS : CCDTURL

Nome abreviado da ferramenta de administração do JMS : CCDT

## Acesso programático

Setters / getters

- MQConnectionFactory.setCCDTURL()
- MQConnectionFactory.getCCDTURL()

## Valores

nulo

Esse é o valor-padrão.

**Um localizador uniforme de recursos (URL)**

## CCSID

Para connection factories, esta propriedade especifica o ID do conjunto de caracteres codificados (CCSID) a ser usado para fluxos de dados internos com o gerenciador de fila... Para destinos, a propriedade define o CCSID a ser usado para codificar dados de sequência em MapMessages, StreamMessages e TextMessages colocados nesse destino.

**Nota:** Normalmente não é necessário alterar essa propriedade para connection factories..

## Objetos Aplicáveis

ConnectionFactory, QueueConnectionFactory, TopicConnectionFactory, Fila, Tópico, XAConnectionFactory, XAQueueConnectionFactory, XATopicConnectionFactory

Nome longo da ferramenta de administração do JMS : CCSID

Nome abreviado da ferramenta de administração do JMS : CCS.

## Acesso programático

Setters / getters

- MQConnectionFactory.setCCSID()
- MQConnectionFactory.getCCSID()

## Valores

**819**

O valor padrão para um connection factory.

**1208**

O valor padrão para um destino.

**Qualquer número inteiro positivo**

**Conceitos relacionados**

[Conversão de mensagens doJMS](#)

## CHANNEL

O nome do canal de conexão do cliente sendo usado.

## Objetos Aplicáveis

ConnectionFactory, QueueConnectionFactory, TopicConnectionFactory, XAConnectionFactory, XAQueueConnectionFactory, XATopicConnectionFactory.

Nome longo da ferramenta de administração do JMS : CHANNEL

Nome abreviado da ferramenta de administração JMS : CHAN

## Acesso programático

Setters / getters

- MQConnectionFactory.setChannel()
- MQConnectionFactory.getChannel()

## Valores

### SYSTEM.DEF.SVRCONN

Esse é o valor-padrão.

### Qualquer sequência válida

## CLEANUP

Nível de Limpeza para Armazenamentos de Assinatura BROKER ou MIGRATE.

## Objetos Aplicáveis

ConnectionFactory, TopicConnectionFactory, XAConnectionFactory, XATopicConnectionFactory

Nome longo da ferramenta de administração do JMS : CLEANUP

Nome abreviado da ferramenta de administração JMS : CL

## Acesso programático

Setters / getters

- MQConnectionFactory.setCleanupNível ()
- MQConnectionFactory.getCleanupNível ()

## Valores

### SEGURO

Use a limpeza segura Esse é o valor-padrão.

### ASPROP

Use limpeza segura, forte ou sem limpeza de acordo com uma propriedade configurada na linha de comando Java .

### Nenhum

Não use nenhuma limpeza

### STRONG

Use limpeza forte.

## CLEANUPINT

O intervalo, em milissegundos, entre execuções de segundo plano do utilitário de limpeza de publicação / assinatura.

## Objetos Aplicáveis

ConnectionFactory, TopicConnectionFactory, XAConnectionFactory, XATopicConnectionFactory

Nome longo da ferramenta de administração do JMS : CLEANUPINT

Nome abreviado da ferramenta de administração JMS : CLINT

## Acesso programático

Setters / getters

- MQConnectionFactory.setCleanupInterval ()
- MQConnectionFactory.getCleanupInterval ()

## Valores

**3600000**

Esse é o valor-padrão.

**Qualquer número inteiro positivo**

## connectionNameList

Lista de nomes de conexões de TCP/IP. A lista é tentada em ordem, uma vez por cada nova tentativa de conexão.

## Objetos Aplicáveis

ConnectionFactory, QueueConnectionFactory, TopicConnectionFactory

Nome longo da ferramenta de administração do JMS : CONNECTIONNAMELIST

Nome abreviado da ferramenta de administração do JMS : CNLIST

## Acesso programático

Setters / getters

- MQConnectionFactory.setconnectionNameList ()
- MQConnectionFactory.getconnectionNameList ()

## Valores

Lista separada por vírgula de HOSTNAME (PORT). HOSTNAME pode ser um nome DNS ou endereço IP.

PORT tem como padrão 1414.

## CLIENTRECONNECTOPTIONS

Opções que governam a reconexão

## Objetos Aplicáveis

ConnectionFactory, QueueConnectionFactory, TopicConnectionFactory

Nome longo da ferramenta de administração do JMS : CLIENTRECONNECTOPTIONS

Nome abreviado da ferramenta de administração do JMS : CROPT

## Acesso programático

Setters / getters

- `MQConnectionFactory.setClientReconnectOptions()`
- `MQConnectionFactory.getClientReconnectOptions()`

## Valores

### QMGR

O aplicativo pode reconectar-se ao mesmo gerenciador de filas ao qual ele está originalmente conectado

Um erro com o código de razão `MQRC_RECONNECT_QMID_MISMATCH` é retornado se o gerenciador de filas ao qual o aplicativo tenta se conectar, conforme especificado na lista de nomes de conexões, tiver um QMID diferente para o gerenciador de filas ao qual ele se conectou originalmente

Use esse valor se um aplicativo puder ser reconectado, mas houver uma afinidade entre o aplicativo IBM MQ classes for JMS e o gerenciador de filas com o qual ele primeiro estabeleceu uma conexão.

Escolha esse valor se desejar que um aplicativo se reconecte automaticamente à instância em espera de um gerenciador de filas altamente disponível.

Para usar esse valor programaticamente, use a constante `WMQConstants.WMQ_CLIENT_RECONNECT_Q_MGR`.

### QUALQUER

O aplicativo pode se reconectar a qualquer um dos gerenciadores de fila especificados na lista de nomes de conexões

Use a opção de reconexão apenas se não houver nenhuma afinidade entre as classes IBM MQ para o aplicativo JMS e o gerenciador de fila com o qual ele estabeleceu inicialmente uma conexão

Para usar esse valor de um programa, use a constante `WMQConstants.WMQ_CLIENT_RECONNECT`.

### DISABLED

O aplicativo não será reconectado.

Para usar esse valor programaticamente, use a constante `WMQConstants.WMQ_CLIENT_RECONNECT_DISABLED`.

### ASDEF

Se o aplicativo será reconectado automaticamente depende do valor do atributo do canal IBM MQ `DefReconnect`.

Esse é o valor-padrão.

Para usar esse valor de um programa, use a constante `WMQConstants.WMQ_CLIENT_RECONNECT_AS_DEF`.

## CLIENTRECONNECTTIMEOUT

Tempo antes do fim das novas tentativas de conexão.

### Objetos Aplicáveis

`ConnectionFactory`, `QueueConnectionFactory`, `TopicConnectionFactory`

Nome longo da ferramenta de administração do JMS : `CLIENTRECONNECTTIMEOUT`

Nome abreviado da ferramenta de administração do JMS : `CRT`.

### Acesso programático

Setters / getters

- `MQConnectionFactory.setClientReconnectTimeout()`

- `MQConnectionFactory.setClientReconnectTimeout()`

## Valores

Intervalo em segundos. Padrão 1800 (30 minutos).

## CLIENTID

O identificador de cliente é utilizado para identificar exclusivamente a conexão do aplicativo para assinaturas duráveis.

### Objetos Aplicáveis

`ConnectionFactory`, `QueueConnectionFactory`, `TopicConnectionFactory`, `XAConnectionFactory`, `XAQueueConnectionFactory`, `XATopicConnectionFactory`.

Nome longo da ferramenta de administração do JMS : CLIENTID

Nome abreviado da ferramenta de administração do JMS : CID

### Acesso programático

Setters / getters

- `MQConnectionFactory.setClientId ()`
- `MQConnectionFactory.getClientId ()`

## Valores

nulo

Esse é o valor-padrão.

**Qualquer sequência válida**

## CLONESUPP

Se duas ou mais instâncias do mesmo assinante de tópico durável puderem ser executadas simultaneamente.

### Objetos Aplicáveis

`ConnectionFactory`, `TopicConnectionFactory`, `XAConnectionFactory`, `XATopicConnectionFactory`

Nome longo da ferramenta de administração do JMS : CLONESUPP

Nome abreviado da ferramenta de administração do JMS : CLS

### Acesso programático

Setters / getters

- `MQConnectionFactory.setCloneSuporte ()`
- `MQConnectionFactory.getCloneSuporte ()`

## Valores

**DISABLED**

Apenas uma instância de um assinante de tópico durável pode ser executada por vez.. Esse é o valor-padrão.

## **ATIVADO**

Duas ou mais instâncias do mesmo assinante de tópico durável podem ser executadas simultaneamente, mas cada instância deve ser executada em uma máquina virtual (JVM) Java separada.

## **COMPHDR**

Uma lista de técnicas que podem ser usadas para compactar os dados do cabeçalho em uma conexão.

### **Objetos Aplicáveis**

ConnectionFactory, TopicConnectionFactory, XAConnectionFactory, XATopicConnectionFactory

Nome longo da ferramenta de administração do JMS : COMPHDR

Nome abreviado da ferramenta de administração do JMS : HC

### **Acesso programático**

Setters / getters

- MQConnectionFactory.setHdrCompList()
- MQConnectionFactory.getHdrCompList()

### **Valores**

#### **Nenhum**

Esse é o valor-padrão.

#### **SISTEMA**

A compactação do cabeçalho da mensagem de RLE é executada

## **COMPMSG**

Uma lista de técnicas que podem ser usadas para compactar os dados da mensagem em uma conexão.

### **Objetos Aplicáveis**

ConnectionFactory, QueueConnectionFactory, TopicConnectionFactory, XAConnectionFactory, XAQueueConnectionFactory, XATopicConnectionFactory.

Nome longo da ferramenta de administração do JMS : COMPMSG

Nome abreviado da ferramenta de administração JMS : MC

### **Acesso programático**

Setters / getters

- MQConnectionFactory.setMsgCompList()
- MQConnectionFactory.getMsgCompList()

### **Valores**

#### **Nenhum**

Esse é o valor-padrão.

**Uma lista de um ou mais dos seguintes valores separados por caracteres em branco:**

RLE ZLIBFAST ZLIBHIGH

## CONNOPT

Controla como os aplicativos do IBM MQ classes for JMS que usam o transporte de ligações se conectam ao gerenciador de filas

### Objetos Aplicáveis

ConnectionFactory, QueueConnectionFactory, TopicConnectionFactory, XAConnectionFactory, XAQueueConnectionFactory, XATopicConnectionFactory.

JMS nome longo da ferramenta de administração: CONNOPT

Nome abreviado da ferramenta de administração do JMS : CNOPT

### Acesso programático

Setters / getters

- MQConnectionFactory.setMQConnectionFactoryOpções ()
- MQConnectionFactory.getMQConnectionFactoryOpções ()

### Valores

#### STANDARD

A natureza da ligação entre o aplicativo e o gerenciador de filas depende do valor do atributo *DefaultBindType* do gerenciador de filas O valor STANDARD é mapeado para IBM MQ *ConnectOption* MQCNO\_STANDARD\_BINDING.

#### SHARED

O aplicativo e o agente do gerenciador de filas locais são executados em unidades separadas de execução, mas compartilham alguns recursos Esse valor é mapeado para o IBM MQ *ConnectOption* MQCNO\_SHARED\_BINDING.

#### isolado

O aplicativo e o agente do gerenciador de filas locais são executados em unidades separadas de execução e não compartilham nenhum recurso O valor ISOLADO é mapeado para o IBM MQ *ConnectOption* MQCNO\_ISOLATED\_BINDING.

#### FASTPATH

O aplicativo e o agente do gerenciador de filas locais são executados na mesma unidade de execução.. Esse valor é mapeado para o IBM MQ *ConnectOption* MQCNO\_FASTPATH\_BINDING.

#### SERIALQM

O aplicativo solicita uso exclusivo da tag de conexão no escopo do gerenciador de filas. Esse valor é mapeado para o IBM MQ *ConnectOption* MQCNO\_SERIALIZE\_CONN\_TAG\_Q\_MGR

#### SERIALQSG

O aplicativo solicita uso exclusivo da tag de conexão no escopo do grupo de filas compartilhadas ao qual o gerenciador de filas pertence. O valor SERIALQSG é mapeado para o IBM MQ *ConnectOption* MQCNO\_SERIALIZE\_CONN\_TAG\_QSG.

#### RESTRICTQM

O aplicativo solicita uso compartilhado da tag de conexão, mas há restrições no uso compartilhado da tag de conexão dentro do escopo do gerenciador de filas. Esse valor é mapeado para o IBM MQ *ConnectOption* MQCNO\_RESTRICT\_CONN\_TAG\_Q\_MGR

#### RESTRICTQSG

O aplicativo solicita uso compartilhado da tag de conexão, mas há restrições no uso compartilhado da tag de conexão dentro do escopo do grupo de filas compartilhadas ao qual o gerenciador de filas pertence. Esse valor é mapeado para o IBM MQ *ConnectOption* MQCNO\_RESTRICT\_CONN\_TAG\_QSG.

Para obter mais informações sobre as opções de conexão do IBM MQ , consulte [Conectando a um gerenciador de filas usando a chamada MQCONN](#).



## CONNTAG

Uma tag que o gerenciador de filas associa aos recursos atualizados pelo aplicativo em uma unidade de trabalho enquanto o aplicativo está conectado ao gerenciador de filas.

### Objetos Aplicáveis

ConnectionFactory, QueueConnectionFactory, TopicConnectionFactory, XAConnectionFactory, XAQueueConnectionFactory, XATopicConnectionFactory.

Nome longo da ferramenta de administração do JMS : CONNTAG

Nome abreviado da ferramenta de administração do JMS : CNTAG.

### Acesso programático

Setters / getters

- MQConnectionFactory.setConnTag ()
- MQConnectionFactory.getConnTag ()

### Valores

**Uma matriz de bytes de 128 elementos, em que cada elemento é 0**

Esse é o valor-padrão.

**Qualquer sequência**

O valor é truncado se for maior que 128 bytes.

## DESCRIÇÃO

Uma descrição do objeto armazenado..

### Objetos Aplicáveis

ConnectionFactory, QueueConnectionFactory, TopicConnectionFactory, Fila, Tópico, XAConnectionFactory, XAQueueConnectionFactory, XATopicConnectionFactory

Nome longo da ferramenta de administração do JMS : DESCRIÇÃO

Nome abreviado da ferramenta de administração do JMS : DESC.

### Acesso programático

Setters / getters

- MQConnectionFactory.setDescription()
- MQConnectionFactory.getDescription()

### Valores

**nulo**

Esse é o valor-padrão.

**Qualquer sequência válida**

## DIRECTAUTH

Se a autenticação do TLS é usada em uma conexão em tempo real com um broker

## Objetos Aplicáveis

ConnectionFactory, TopicConnectionFactory

Nome longo da ferramenta de administração do JMS : DIRECTAUTH

Nome abreviado da ferramenta de administração JMS : DAUTH

## Acesso programático

Setters / getters

- MQConnectionFactory.setDirectAuth ()
- MQConnectionFactory.getDirectAuth ()

## Valores

### BÁSICA

Nenhuma autenticação, autenticação de nome de usuário ou autenticação de senha. Esse é o valor-padrão.

### Certificado

Autenticação de certificado de chave pública

## ENCODING

Como os dados numéricos no corpo de uma mensagem são representados quando a mensagem é enviada para esse destino A propriedade especifica a representação de números inteiros binários, números inteiros decimais empacotados e números de vírgula flutuante..

## Objetos Aplicáveis

Fila, Tópico

Nome longo da ferramenta de administração do JMS : ENCODING

Nome abreviado da ferramenta de administração do JMS : ENC.

## Acesso programático

Setters / getters

- MQDestination.setEncoding()
- MQDestination.getEncoding()

## Valores

### Propriedade Encoding

Os valores válidos que a propriedade ENCODING pode tomar são construídos a partir das três subpropriedades:

#### Codificação de Inteiro

Normal ou invertido

#### Codificação Decimal

Normal ou invertido

#### codificação de vírgula flutuante

IEEE normal, IEEE revertido ou z/OS

A propriedade ENCODING é expressa como uma sequência de três caracteres com a sintaxe a seguir:

```
{N|R}{N|R}{N|R|3}
```

Nesta cadeia:

- N denota normal
- R denota reverso
- 3 denota z/OS
- O primeiro caractere representa *codificação de número inteiro*
- O segundo caractere representa *codificação decimal*
- O terceiro caractere representa a *codificação de vírgula flutuante*

Isso fornece um conjunto de doze valores possíveis para a propriedade `ENCODING`

Há um valor adicional, a sequência `NATIVE`, que configura valores de codificação apropriados para a plataforma Java .

Os exemplos a seguir mostram combinações válidas para `ENCODING`:

```
ENCODING (NNR)
ENCODING (NATIVE)
ENCODING (RR3)
```

## EXPIRY

O tempo após o qual as mensagens em um destino expiram

### Objetos Aplicáveis

Fila, Tópico

Nome longo da ferramenta de administração do JMS : EXPIRAÇÃO

Nome abreviado da ferramenta de administração do JMS : EXP.

### Acesso programático

Setters / getters

- `MQDestination.setExpiry()`
- `MQDestination.getExpiry()`

### Valores

#### AP

A expiração pode ser definida pelo aplicativo JMS Esse é o valor-padrão.

#### UNLIM

Não ocorre nenhuma expiração

#### 0

Não ocorre nenhuma expiração

**Qualquer número inteiro positivo representando expiração em milissegundos.**

## FAILIFQUIESCE

Esta propriedade determina se as chamadas para determinados métodos falharão se o gerenciador de filas estiver em um estado quiesce ou se um aplicativo estiver se conectando a um gerenciador de filas usando o transporte `CLIENT` e o canal que o aplicativo está usando foi colocado em um estado quiesce, por exemplo, usando o comando **STOP CHANNEL** ou **STOP CHANNEL MODE(QUIESCE)** `MQSC`.

## Objetos Aplicáveis

ConnectionFactory, QueueConnectionFactory, TopicConnectionFactory, Fila, Tópico, XAConnectionFactory, XAQueueConnectionFactory, XATopicConnectionFactory

Nome longo da ferramenta de administração JMS : FAILIFQUIESCE

Nome abreviado da ferramenta de administração do JMS : FIQ

## Acesso programático

Setters / getters

- MQConnectionFactory.setFailIfQuiesce()
- MQConnectionFactory.getFailIfQuiesce()

## Valores

### SIM

Chamadas para determinados métodos falharão se o gerenciador de filas estiver em um estado de quiesce ou o canal que está sendo usado para se conectar a um gerenciador de filas estiver em quiesce. Se um aplicativo detectar qualquer uma dessas condições, o aplicativo poderá concluir sua tarefa imediata e fechar a conexão, permitindo que o gerenciador de filas ou a instância do canal pare. Esse é o valor-padrão.

### NÃO

Nenhuma chamada de método falha porque o gerenciador de filas ou o canal que está sendo usado para se conectar a um gerenciador de filas está em um estado de quiesce. Se você especificar esse valor, um aplicativo não poderá detectar que o gerenciador ou canal de filas está quiesce. O aplicativo pode continuar a executar operações no gerenciador de filas e, portanto, evitar que o gerenciador de filas pare.

## HOSTNAME

Para uma conexão com um gerenciador de filas, o nome do host ou endereço IP do sistema no qual o gerenciador de filas está em execução ou, para uma conexão em tempo real com um broker, o nome do host ou endereço IP do sistema no qual o broker está em execução.

## Objetos Aplicáveis

ConnectionFactory, QueueConnectionFactory, TopicConnectionFactory, XAConnectionFactory, XAQueueConnectionFactory, XATopicConnectionFactory.

Nome longo da ferramenta de administração do JMS : HOSTNAME

Nome abreviado da ferramenta de administração do JMS : HOST

## Acesso programático

Setters / getters

- MQConnectionFactory.setHostName ()
- MQConnectionFactory.getHostName ()

## Valores

### host local

Esse é o valor-padrão.

### Qualquer sequência válida

## LOCALADDRESS

Para uma conexão com um gerenciador de filas, essa propriedade especifica a interface de rede local a ser usada ou a porta local ou o intervalo de portas locais a ser usado.

### Objetos Aplicáveis

ConnectionFactory, QueueConnectionFactory, TopicConnectionFactory, XAConnectionFactory, XAQueueConnectionFactory, XATopicConnectionFactory.

Nome longo da ferramenta de administração do JMS : LOCALADDRESS

Nome abreviado da ferramenta de administração do JMS : LA.

### Acesso programático

Setters / getters

- MQConnectionFactory.setLocalAddress ()
- MQConnectionFactory.getLocalAddress ()

### Valores

#### "" (sequência vazia)

Esse é o valor-padrão.

#### Uma sequência no formato [ip-addr] [(low-port [, high-port])]

Estes são alguns exemplos:

192.0.2.0

O canal se liga ao endereço 192.0.2.0 localmente.

192.0.2.0(1000)

O canal se liga ao endereço 192.0.2.0 localmente e usa a porta 1000.

192.0.2.0(1000,2000)

O canal se liga ao endereço 192.0.2.0 localmente e usa uma porta no intervalo de 1000 a 2000.

(1000)

O canal se liga à porta 1000 localmente.

(1000,2000)

O canal se liga a uma porta no intervalo de 1000 a 2000 localmente.

É possível especificar um nome do host, em vez de um endereço IP Para uma conexão em tempo real com um broker, essa propriedade é relevante apenas quando multicast é usado e o valor da propriedade não deve conter um número de porta ou um intervalo de números de porta. Os únicos valores válidos da propriedade neste caso são null, um endereço IP ou um nome do host.

## MAPNAMESTYLE

Permite que o estilo de compatibilidade seja usado para os nomes de elementos MapMessage

### Objetos Aplicáveis

ConnectionFactory, QueueConnectionFactory, TopicConnectionFactory, XAConnectionFactory, XAQueueConnectionFactory, XATopicConnectionFactory.

Nome longo da ferramenta de administração do JMS : MAPNAMESTYLE

Nome abreviado da ferramenta de administração do JMS : MNST

## Acesso programático

Setters / getters

- MQConnectionFactory.setMapNameStyle()
- MQConnectionFactory.getMapNameStyle()

## Valores

### STANDARD

O formato de nomenclatura do elemento com.ibm.jms.JMSMapMessage padrão deve ser usado. Esse é o valor padrão e permite que identificadores não legais do Java sejam usados como o nome do elemento.

### Compatível

O formato de nomenclatura do elemento com.ibm.jms.JMSMapMessage mais antigo deve ser usado. Somente identificadores Java legais podem ser usados como o nome do elemento. Isso será necessário apenas se as mensagens de mapa estiverem sendo enviadas para um aplicativo que esteja usando uma versão de IBM MQ classes for JMS anterior a 5.3.

## MAXBUFFSIZE

O número máximo de mensagens recebidas que podem ser armazenadas em um buffer de mensagem interno enquanto aguardam para ser processada pelo aplicativo. Essa propriedade se aplica apenas quando TRANSPORT possui o valor DIRECT ou DIRECTHTTP..

## Objetos Aplicáveis

ConnectionFactory, TopicConnectionFactory

Nome longo da ferramenta de administração do JMS : MAXBUFFSIZE

Nome abreviado da ferramenta de administração do JMS : MBSZ

## Acesso programático

Setters / getters

- MQConnectionFactory.setMaxBufferSize()
- MQConnectionFactory.getMaxBufferSize()

## Valores

### 1000

Esse é o valor-padrão.

**Qualquer número inteiro positivo**

## MDREAD

Essa propriedade determina se um aplicativo JMS pode extrair os valores dos campos do MQMD.

## Objetos Aplicáveis

Nome longo da ferramenta de administração do JMS : MDREAD

Nome abreviado da ferramenta de administração do JMS : MDR

## Acesso programático

Setters / getters

- MQDestination.setMQMDReadEnabled()
- MQDestination.getMQMDReadEnabled()

## Valores

### NÃO

Ao enviar mensagens, as propriedades JMS\_IBM\_MQMD\* em uma mensagem enviada não são atualizadas para refletir os valores de campos atualizados no MQMD. Ao receber mensagens, nenhuma das propriedades JMS\_IBM\_MQMD\* estarão disponíveis em uma mensagem recebida, mesmo que o emissor tenha configurado algumas ou todas elas. Este é o valor padrão para ferramentas administrativas.

Para os programas, use False.

### Sim

Ao enviar mensagens, todas as propriedades JMS\_IBM\_MQMD\* em uma mensagem enviada são atualizadas para refletir os valores de campo atualizados no MQMD, incluindo as propriedades que o emissor não configurou explicitamente. Ao receber mensagens, todas as propriedades JMS\_IBM\_MQMD\* estão disponíveis em uma mensagem recebida, incluindo as propriedades que o emissor não configurou explicitamente.

Para programas, use Verdadeiro.

## MDWRITE

Essa propriedade determina se um aplicativo JMS pode configurar os valores de campos do MQMD

### Objetos Aplicáveis

Fila, Tópico

Nome longo da ferramenta de administração do JMS : MDWRITE

Nome abreviado da ferramenta de administração do JMS : MDR

### Acesso programático

Setters / getters

- MQDestination.setMQMDWriteEnabled()
- MQDestination.getMQMDWriteEnabled()

## Valores

### NÃO

Todas as propriedades JMS\_IBM\_MQMD\* são ignoradas e seus valores não são copiados para a estrutura MQMD subjacente. Este é o valor padrão para ferramentas administrativas.

Para os programas, use False.

### SIM

As propriedades JMS\_IBM\_MQMD\* são processadas. Seus valores serão copiados para a estrutura do MQMD subjacente.

Para programas, use Verdadeiro.

## MDMSGCTX

Qual nível de contexto da mensagem deve ser configurado pelo aplicativo JMS. O aplicativo deve estar em execução com autoridade de contexto apropriado para esta propriedade entrar em vigor.

## Objetos Aplicáveis

Nome longo da ferramenta de administração do JMS : MDMSGCTX

Nome abreviado da ferramenta de administração do JMS : MDCTX

## Acesso programático

Setters / getters

- MQDestination.setMQMDMessageContext()
- MQDestination.getMQMDMessageContext()

## Valores

### DEFAULT

A chamada API MQOPEN e a estrutura MQPMO não especificam nenhuma opção de contexto de mensagem explícita. Este é o valor padrão para ferramentas administrativas.

Para programas, use WMQ\_MDCTX\_DEFAULT..

### SET\_IDENTITY\_CONTEXT

A chamada da API MQOPEN especifica a opção de contexto da mensagem

MQOO\_SET\_IDENTITY\_CONTEXT e a estrutura MQPMO especifica MQPMO\_SET\_IDENTITY\_CONTEXT

Para programas, use WMQ\_MDCTX\_SET\_IDENTITY\_CONTEXT..

### SET\_ALL\_CONTEXT

A chamada da API MQOPEN especifica a opção de contexto da mensagem MQOO\_SET\_ALL\_CONTEXT e a estrutura MQPMO especifica MQPMO\_SET\_ALL\_CONTEXT

Para programas, use WMQ\_MDCTX\_SET\_ALL\_CONTEXT..

## MSGBATCHSZ

O número máximo de mensagens a serem obtidas de uma fila em um pacote ao usar a entrega de mensagem assíncrona.

## Objetos Aplicáveis

ConnectionFactory, QueueConnectionFactory, TopicConnectionFactory, XAConnectionFactory, XAQueueConnectionFactory, XATopicConnectionFactory.

Nome longo da ferramenta de administração do JMS : MAXBUFFSIZE

Nome abreviado da ferramenta de administração do JMS : MBSZ

## Acesso programático

Setters / getters

- MQConnectionFactory.setMsgBatchSize()
- MQConnectionFactory.getMsgBatchSize()

## Valores

### 10

Esse é o valor-padrão.

**Qualquer número inteiro positivo**



## MSGBODY

Determina se um aplicativo JMS acessa o MQRFH2 de uma mensagem IBM MQ como parte da carga útil da mensagem.

### Objetos Aplicáveis

Fila, Tópico

Nome longo da ferramenta de administração do JMS : WMQ\_MESSAGE\_BODY

Nome abreviado da ferramenta de administração JMS : MBODY

### Acesso programático

Setters / getters

- MQConnectionFactory.setMessageBodyStyle()
- MQConnectionFactory.getMessageBodyStyle()

### Valores

#### UNSPECIFIED

Ao enviar, o IBM MQ classes for JMS gera e inclui um cabeçalho MQRFH2, ou não, dependendo do valor de WMQ\_TARGET\_CLIENT. Ao receber, age como valor JMS.

#### JMS

Ao enviar, o IBM MQ classes for JMS gera automaticamente um cabeçalho MQRFH2 e o inclui na mensagem do IBM MQ.

Ao receber, o IBM MQ classes for JMS configura as propriedades de mensagem do JMS de acordo com os valores no MQRFH2 (se presente); ele não apresenta o MQRFH2 como parte do corpo da mensagem do JMS.

#### MQ

Ao enviar, o IBM MQ classes for JMS não gera um MQRFH2.

Ao receber, o IBM MQ classes for JMS apresenta o MQRFH2 como parte do corpo da mensagem do JMS.

## MSGRETENTION

Se o consumidor de conexão mantém as mensagens não entregues na fila de entrada

### Objetos Aplicáveis

ConnectionFactory, QueueConnectionFactory, XAConnectionFactory, XAQueueConnectionFactory,

Nome longo da ferramenta de administração do JMS : MSGRETENTION

Nome abreviado da ferramenta de administração do JMS : MRET.

### Acesso programático

Setters / getters

- MQConnectionFactory.setMensagemRetenção ()
- MQConnectionFactory.getMensagemRetenção ()

### Valores

#### Sim

As mensagens não entregues permanecem na fila de entrada. Esse é o valor-padrão.

## No

Mensagens não entregues são tratadas de acordo com suas opções de disposição.

## MSGSELECTION

Determina se a seleção de mensagens é feita pelo IBM MQ classes for JMS ou pelo broker. Se TRANSPORT tiver o valor DIRECT, a seleção de mensagens será sempre feita pelo broker e o valor de MSGSELECTION será ignorado.. A seleção de mensagens pelo broker não é suportada quando BROKERVER tem o valor V1.

### Objetos Aplicáveis

ConnectionFactory, TopicConnectionFactory, XAConnectionFactory, XATopicConnectionFactory

Nome longo da ferramenta de administração do JMS : MSGSELECTION

Nome abreviado da ferramenta de administração do JMS : MSEL

### Acesso programático

Setters / getters

- MQConnectionFactory.setMessageSelection ()
- MQConnectionFactory.getMessageSelection ()

### Valores

#### CLIENTE

A seleção de mensagens é feita por IBM MQ classes for JMS. Esse é o valor-padrão.

#### BROKER

A seleção de mensagens é feita pelo broker

## MULTICAST

Para ativar multicast em uma conexão em tempo real com um broker e, se ativado, para especificar a maneira precisa na qual multicast é usado para entregar mensagens do broker para um consumidor de mensagens. A propriedade não tem efeito sobre como um produtor de mensagem envia mensagens para um broker

### Objetos Aplicáveis

ConnectionFactory, TopicConnectionFactory, Tópico

Nome longo da ferramenta de administração do JMS : MULTICAST

Nome abreviado da ferramenta de administração do JMS : MCAST

### Acesso programático

Setters / getters

- MQConnectionFactory.setMulticast()
- MQConnectionFactory.getMulticast()

### Valores

#### DISABLED

As mensagens não são entregues a um consumidor de mensagens usando transporte multicast. Esse é o valor padrão para os objetos de Factory ConnectionFactory e TopicConnection

### **ASCF**

As mensagens são entregues a um consumidor de mensagens de acordo com a configuração de multicast para o connection factory associado ao consumidor de mensagem. A configuração de multicast para o connection factory é observada no momento em que o consumidor de mensagem é criado. Esse valor é válido apenas para objetos Topic e é o valor padrão para objetos Topic..

### **ATIVADO**

Se o tópico for configurado para multicast no broker, as mensagens serão entregues a um consumidor de mensagem usando o transporte multicast. Uma qualidade de serviço confiável será usada, se o tópico for configurado para multicast confiável.

### **RELIABLE**

Se o tópico for configurado para multicast confiável no broker, as mensagens serão entregues ao consumidor de mensagem usando transporte multicast com uma qualidade de serviço confiável. Se o tópico não estiver configurado para multicast confiável, não será possível criar um consumidor de mensagens para o tópico.

### **NOTR**

Se o tópico for configurado para multicast no broker, as mensagens serão entregues para o consumidor de mensagens usando o transporte multicast. Uma qualidade de serviço confiável não será usada, mesmo se o tópico estiver configurado para multicast confiável.

## **OPTIMISTICPUBLICATION**

Essa propriedade determina se o IBM MQ classes for JMS retorna o controle imediatamente para um publicador que publicou uma mensagem ou se ele retorna o controle somente após ter concluído todo o processamento associado à chamada e pode relatar o resultado para o publicador.

### **Objetos Aplicáveis**

ConnectionFactory, TopicConnectionFactory

JMS nome longo da ferramenta de administração: OPTIMISTICPUBLICATION

Nome abreviado da ferramenta de administração do JMS : OPTPUB

### **Acesso programático**

Setters / getters

- MQConnectionFactory.setOptimisticPublication ()
- MQConnectionFactory.getOptimisticPublication ()

### **Valores**

#### **NÃO**

Quando um publicador publica uma mensagem, IBM MQ classes for JMS não retorna o controle para o publicador até que ele tenha concluído todo o processamento associado à chamada e possa relatar o resultado para o publicador. Esse é o valor-padrão.

#### **SIM**

Quando um publicador publica uma mensagem, o IBM MQ classes for JMS retorna o controle para o publicador imediatamente, antes de concluir todo o processamento associado à chamada e pode relatar o resultado para o publicador. IBM MQ classes for JMS relata o resultado apenas quando o publicador confirma a mensagem..

## **OUTCOMENOTIFICATION**

Essa propriedade determina se o IBM MQ classes for JMS retorna o controle imediatamente para um assinante que acabou de confirmar ou confirmar uma mensagem ou se ele retorna o controle somente após ter concluído todo o processamento associado à chamada e pode relatar o resultado para o assinante.

## Objetos Aplicáveis

ConnectionFactory, TopicConnectionFactory

Nome longo da ferramenta de administração do JMS : OUTCOMENOfício

Nome abreviado da ferramenta de administração do JMS : NOTIFY

## Acesso programático

Setters / getters

- MQConnectionFactory.setOutcomeNotification ()
- MQConnectionFactory.getOutcomeNotification ()

## Valores

### SIM

Quando um assinante reconhece ou confirma uma mensagem, IBM MQ classes for JMS não retorna o controle para o assinante até que ele tenha concluído todo o processamento associado à chamada e possa relatar o resultado para o assinante. Esse é o valor-padrão.

### NÃO

Quando um assinante reconhece ou confirma uma mensagem, o IBM MQ classes for JMS retorna o controle para o assinante imediatamente, antes de ter concluído todo o processamento associado à chamada e pode relatar o resultado para o assinante.

## PERSISTENCE

A persistência de mensagens enviadas para um destino

## Objetos Aplicáveis

Fila, Tópico

Nome longo da ferramenta de administração do JMS : PERSISTENCE

Nome abreviado da ferramenta de administração JMS : PER

## Acesso programático

Setters / getters

- MQDestination.setPersistence()
- MQDestination.getPersistence()

## Valores

### AP

A persistência é definida pelo aplicativo JMS. Esse é o valor-padrão.

### qdef

A persistência assume o valor padrão da fila.

### pers

As mensagens são persistentes

### NÃO

As mensagens não são persistentes

### HIGH

Consulte [JMS mensagens persistentes](#) para obter informações adicionais sobre o uso desse valor.

## POLLINGINT

Se cada listener de mensagem em uma sessão não tiver nenhuma mensagem adequada em sua fila, este será o intervalo máximo, em milissegundos, que decorre antes de cada listener de mensagem tentar novamente obter uma mensagem de sua fila. Se ocorrer com frequência o fato de nenhuma mensagem adequada estar disponível para qualquer um dos listeners da mensagem em uma sessão, considere aumentar o valor desta propriedade. Esta propriedade é relevante apenas se TRANSPORT tiver o valor BIND ou CLIENT.

### Objetos Aplicáveis

ConnectionFactory, QueueConnectionFactory, TopicConnectionFactory, XAConnectionFactory, XAQueueConnectionFactory, XATopicConnectionFactory.

Nome longo da ferramenta de administração do JMS : POLLINGINT

Nome abreviado da ferramenta de administração JMS : PINT

### Acesso programático

Setters / getters

- MQConnectionFactory.setPollingInterval ()
- MQConnectionFactory.getPollingInterval ()

### Valores

#### 5000

Esse é o valor-padrão.

**Qualquer número inteiro positivo**

## PORT

Para uma conexão com um gerenciador de filas, o número da porta na qual o gerenciador de filas está atendendo ou, para uma conexão em tempo real com um broker, o número da porta na qual o broker está atendendo conexões em tempo real..

### Objetos Aplicáveis

ConnectionFactory, QueueConnectionFactory, TopicConnectionFactory, XAConnectionFactory, XAQueueConnectionFactory, XATopicConnectionFactory.

Nome longo da ferramenta de administração do JMS : PORT

Nome abreviado da ferramenta de administração do JMS : PORT

### Acesso programático

Setters / getters

- MQConnectionFactory.setPort()
- MQConnectionFactory.getPort()

### Valores

#### 1414

Esse será o valor padrão se TRANSPORT estiver configurado como CLIENT.

#### 1506

Este é o valor padrão se TRANSPORT estiver configurado como DIRECT ou DIRECTHTTP...

**Qualquer número inteiro positivo**

## **PRIORITY**

A prioridade para as mensagens enviadas para um destino.

### **Objetos Aplicáveis**

Fila, Tópico

Nome longo da ferramenta de administração do JMS : PRIORITY

Nome abreviado da ferramenta de administração do JMS : PRI

### **Acesso programático**

Setters / getters

- MQDestination.setPriority()
- MQDestination.getPriority()

### **Valores**

#### **AP**

Prioridade é definida pelo aplicativo JMS . Esse é o valor-padrão.

#### **qdef**

Priority assume o valor do padrão da fila.

#### **Qualquer número inteiro no intervalo de 0 a 9**

Do mais baixo ao mais alto.

## **PROCESSDURATION**

Essa propriedade determina se um assinante garante processar rapidamente qualquer mensagem recebida antes de retornar o controle para IBM MQ classes for JMS.

### **Objetos Aplicáveis**

ConnectionFactory, TopicConnectionFactory

Nome longo da ferramenta de administração do JMS : PROCESSDURATION

Nome abreviado da ferramenta de administração do JMS : PROCDUR.

### **Acesso programático**

Setters / getters

- MQConnectionFactory.setProcessDuração ().
- MQConnectionFactory.getProcessDuração ()

### **Valores**

#### **DESCONHECIDO**

Um assinante não pode dar nenhuma garantia sobre a rapidez com que pode processar qualquer mensagem que recebe. Esse é o valor-padrão.

#### **CURTO**

Um assinante garante processar rapidamente qualquer mensagem recebida antes de retornar o controle para IBM MQ classes for JMS.

## PROVIDERVERSION

Essa propriedade diferencia entre os três modos de operação do sistema de mensagens do IBM MQ : IBM MQ modo normal do provedor de sistemas de mensagens, IBM MQ modo normal do provedor de sistemas de mensagens com restrições e IBM MQ modo de migração do provedor de sistemas de mensagens.

O modo normal do provedor de mensagens IBM MQ usa todos os recursos de um gerenciador de filas IBM MQ para implementar JMS. Esse modo é otimizado para usar a API e a funcionalidade do JMS 2.0. O modo normal do provedor de sistemas de mensagens do IBM MQ com restrições usa a API JMS 2.0 , mas não os novos recursos como assinaturas compartilhadas, entrega atrasada ou envio assíncrono.

### Objetos Aplicáveis

ConnectionFactory, QueueConnectionFactory, TopicConnectionFactory, XAConnectionFactory, XAQueueConnectionFactory, XATopicConnectionFactory.

Nome longo da ferramenta de administração do JMS : PROVIDERVERSION

Nome abreviado da ferramenta de administração do JMS : PVER

### Acesso programático

Setters / getters

- MQConnectionFactory.setProviderVersão ()
- MQConnectionFactory.getProviderVersão ()

### Valores

É possível configurar a propriedade **PROVIDERVERSION** para qualquer um dos valores 8 (modo normal), 7 (modo normal com restrições), 6 (modo de migração) ou não especificado (o valor padrão). O valor que você especificar para a propriedade **PROVIDERVERSION** precisa ser uma sequência. Se você estiver especificando uma opção 8, 7 ou 6, poderá fazer isso em qualquer um dos seguintes formatos:

- V.R.M.F
- V.R.M
- V.R
- V

em que V, R, M e F são valores inteiros maiores ou iguais a zero. Os valores R, M e F extras são opcionais e estão disponíveis para você usar em controle de granularidade fina, caso seja necessário. Por exemplo, caso quisesse usar um nível **PROVIDERVERSION** de 7, a configuração seria **PROVIDERVERSION=7, 7.0, 7.0.0** ou **7.0.0.0**.

#### 8 - Modo normal

O aplicativo JMS usa o modo normal do provedor de sistemas de mensagens do IBM MQ. O modo normal usa todos os recursos de um gerenciador de filas do IBM MQ para implementar o JMS. Esse modo é otimizado para usar a API e a funcionalidade do JMS 2.0.

Se você estiver se conectando a um gerenciador de fila com um nível de comando de 800 ou mais recente, todos os recursos e API do JMS 2.0 , como envio assíncrono, entrega atrasada ou assinatura compartilhada, poderão ser usados.

Se o gerenciador de filas especificado nas configurações de connection factory não for um gerenciador de filas IBM MQ 8.0.0 ou posterior, o método `createConnection` falhará com a exceção `JMSFMQ0003`

O modo normal provedor de mensagens IBM MQ usa o recurso de conversas de compartilhamento e o número de conversas que podem ser compartilhadas é controlado pela propriedade **SHARECNV()** no canal de conexão do servidor. Se essa propriedade for configurada como 0, não será possível usar

o modo normal do provedor de sistemas de mensagens do IBM MQ e o método `createConnection` falhará com uma exceção `JMSCC5007`.

### 7 - Modo normal com restrições

O aplicativo JMS usa o modo normal do provedor de sistemas de mensagens do IBM MQ com restrições. Esse modo usa a API do JMS 2.0, mas não os novos recursos, como assinaturas compartilhadas, entrega atrasada ou envio assíncrono.

Ao configurar **PROVIDERVERSION** como 7, apenas o provedor de mensagens IBM MQ normal.

Se você estiver conectando usando o modo normal com restrições, a um gerenciador de filas com um nível de comando inferior a 800, será possível usar a API JMS 2.0, mas não os recursos de envio assíncrono, entrega atrasada ou assinatura compartilhada.

O modo normal do provedor de mensagens IBM MQ com restrições, que usa o recurso de conversas de compartilhamento e o número de conversas que podem ser compartilhadas, é controlado pela propriedade **SHARECNV** () no canal de conexão do servidor. Se essa propriedade for configurada como 0, não será possível usar o modo normal do provedor de sistemas de mensagens do IBM MQ com restrições e o método `createConnection` falhará com uma exceção `JMSCC5007`.

### 6 - Modo de Migração

O aplicativo JMS utiliza o modo de migração do provedor de sistema de mensagens do IBM MQ.

É possível se conectar a um gerenciador de filas do IBM MQ 8.0 ou mais recente usando esse modo, mas nenhum dos novos recursos de um gerenciador de filas do IBM MQ classes for JMS é usado, por exemplo, leia adiante ou fluxo.

Se você tiver um cliente IBM MQ 8.0 ou mais recente se conectando a um gerenciador de filas do IBM MQ 8.0 ou mais recente, a seleção de mensagem será feita pelo gerenciador de filas em vez do sistema do cliente.

Se o modo de migração do provedor de sistemas de mensagens do IBM MQ for especificado e você tentar usar qualquer API do JMS 2.0, a chamada de método de API falhará com a exceção `JMSCC5007`.

### unspecified (padrão)

A propriedade **PROVIDERVERSION** é definida como *não especificada* por padrão.

Um connection factory que foi criado com uma versão anterior do IBM MQ classes for JMS no JNDI obtém esse valor quando o connection factory é usado com a nova versão do IBM MQ classes for JMS. O seguinte algoritmo é usado para determinar o modo de operação que é usado. Este algoritmo é usado quando o método `createConnection` é chamado e usa outros aspectos do connection factory para determinar se o modo normal do provedor de sistema de mensagens do IBM MQ, o modo normal com restrições ou o modo de migração do provedor de mensagens do IBM MQ é necessário.

1. Primeiro, uma tentativa de usar o modo normal do provedor de sistema de mensagens do IBM MQ é feita.
2. Se o gerenciador de filas conectado não for o IBM MQ 8.0 ou mais recente, será feita uma tentativa de usar o modo normal do provedor de sistemas de mensagens do IBM MQ com restrições.
3. Se o gerenciador de filas conectado não for o IBM WebSphere MQ 7.0.1 ou mais recente, a conexão será encerrada e o modo de migração do provedor de sistemas de mensagens do IBM MQ será usado como alternativa.
4. Se a propriedade **SHARECNV** no canal de conexão do servidor for definida como 0, a conexão será fechada, e a partir daí será usado o modo de migração do provedor de mensagens do IBM MQ.
5. Se **BROKERVER** for configurado como V1 ou o valor padrão *não especificado*, IBM MQ o modo normal do provedor de sistemas de mensagens continuará a ser usado..

Consulte [ALTER QMGR](#) para obter informações sobre o parâmetro PSMODE do comando ALTER QMGR para obter informações adicionais sobre compatibilidade.

6. Se o **BROKERVER** for configurado como V2, a ação realizada dependerá do valor de **BROKERQMGR** :
  - Se o **BROKERQMGR** estiver em branco:



Se a fila especificada pela propriedade **BROKERCONQ** puder ser aberta para saída (ou seja, o MQOPEN para saída for bem-sucedido) e **PSMODE** no gerenciador de filas for configurado como COMPAT ou DISABLED, o modo de migração do provedor de sistemas de mensagens do IBM MQ será usado.

- Se a fila especificada pela propriedade **BROKERCONQ** não puder ser aberta para saída, ou o atributo **PSMODE** estiver configurado como ATIVADO:

O modo normal do provedor de sistemas de mensagens do IBM MQ será usado.

- Se **BROKERQMGR** não estiver em branco:

O modo de migração do provedor de sistemas de mensagens do IBM MQ será usado.

Se não for possível mudar o Connection Factory que você está usando, substitua qualquer configuração usando a propriedade `com.ibm.msg.client.wmq.overrideProviderVersion` no Connection Factory. Esta substituição se aplica a todos os connection factories na JVM, mas os objetos connection factory da conexão real não serão modificados.

### Tarefas relacionadas

Configurando a propriedade JMS **PROVIDERVERSION**

## PROXYHOSTNAME

O nome do host ou endereço IP do sistema no qual o servidor proxy está em execução ao usar uma conexão em tempo real com um broker por um servidor proxy.

### Objetos Aplicáveis

ConnectionFactory, TopicConnectionFactory

Nome longo da ferramenta de administração do JMS : PROXYHOSTNAME

Nome abreviado da ferramenta de administração do JMS : PHOST.

### Acesso programático

Setters / getters

- MQConnectionFactory.setProxyHostName()
- MQConnectionFactory.getProxyHostName()

### Valores

nulo

O nome do host do servidor proxy. Esse é o valor-padrão.

## PROXYPORT

O número da porta na qual o servidor proxy está atendendo ao usar uma conexão em tempo real com um broker por meio de um servidor proxy

### Objetos Aplicáveis

ConnectionFactory, TopicConnectionFactory

Nome longo da ferramenta de administração do JMS : PROXYPORT

Nome abreviado da ferramenta de administração do JMS : PPORT

### Acesso programático

Setters / getters

MQConnectionFactory.setProxyPorta ()

MQConnectionFactory.getProxyPorta ()

## Valores

443

O número da porta do servidor proxy. Esse é o valor-padrão.

## PUBACKINT

O número de mensagens publicadas por um publicador antes de o IBM MQ classes for JMS solicitar uma confirmação do broker.

Quando você diminui o valor dessa propriedade, o IBM MQ classes for JMS solicita confirmações com mais frequência, portanto, o desempenho do publicador diminui. Quando você aumentar o valor, IBM MQ classes for JMS levará mais tempo para lançar uma exceção se o broker falhar. Esta propriedade é relevante apenas se TRANSPORT tiver o valor BIND ou CLIENT.

## Objetos Aplicáveis

ConnectionFactory, TopicConnectionFactory, XAConnectionFactory, XATopicConnectionFactory

Nome longo da ferramenta de administração do JMS : PROXYPORT

Nome abreviado da ferramenta de administração do JMS : PPORT

## Acesso programático

Setters / getters

MQConnectionFactory.setPubAckInterval()

MQConnectionFactory.getPubAckInterval()

## Valores

25

Qualquer número inteiro positivo pode ser o valor padrão.

## PUTASYNCALLOWED

Essa propriedade determina se os produtores de mensagens têm permissão para usar as postagens assíncronas para enviar mensagens para esse destino.

## Objetos Aplicáveis

Fila, Tópico

Nome longo da ferramenta de administração do JMS : PUTASYNCALLOWED

Nome abreviado da ferramenta de administração do JMS : PAALD.

## Acesso programático

Setters / getters

MQDestination.setPutAsyncAllowed()

MQDestination.getPutAsyncAllowed()

## Valores

### AS\_DEST

Determine se as colocações assíncronas são permitidas consultando a definição de fila ou tópico. Esse é o valor-padrão.

### AS\_Q\_DEF

Determine se as entradas assíncronas são permitidas consultando a definição de fila.

### AS\_TOPIC\_DEF

Determine se as colocações assíncronas são permitidas referindo-se à definição de tópico

### NÃO

Puts assíncronos não são permitidos.

### SIM

As colocações assíncronas são permitidas

## QMANAGER

O nome do gerenciador de filas para conexão.

No entanto, se seu aplicativo usar uma tabela de definição de canal do cliente para se conectar a um gerenciador de filas, consulte [Usando uma tabela de definição de canal do cliente com IBM MQ classes for JMS](#).

### Objetos Aplicáveis

ConnectionFactory, QueueConnectionFactory, TopicConnectionFactory, Fila, XAConnectionFactory, XAQueueConnectionFactory, XATopicConnectionFactory

Nome longo da ferramenta de administração do JMS : QMANAGER

Nome abreviado da ferramenta de administração do JMS : QMGR

### Acesso programático

Setters / getters

- MQConnectionFactory.setQueueManager ()
- MQConnectionFactory.getQueueManager ()

## Valores

### "" (sequência vazia)

Qualquer sequência pode ser o valor padrão.

## FILA

O nome do destino de fila do JMS Isso corresponde ao nome da fila usada pelo gerenciador de filas

### Objetos Aplicáveis

Fila

Nome longo da ferramenta de administração do JMS : QUEUE

Nome abreviado da ferramenta de administração do JMS : QU

## Valores

### Qualquer sequência

Qualquer nome de fila IBM MQ válido.

## Referências relacionadas

[Regras para nomear objetos IBM MQ >](#)

## READAHEADALLOWED

Essa propriedade determina se os consumidores de mensagens e navegadores de fila têm permissão para usar leitura antecipada para obter mensagens não persistentes desse destino em um buffer interno antes de recebê-las.

### Objetos Aplicáveis

Fila, Tópico

Nome longo da ferramenta de administração do JMS : READAHEADALLOWED

Nome abreviado da ferramenta de administração do JMS : RAALD

### Acesso programático

Setters / getters

- MQDestination.setReadAheadAllowed()
- MQDestination.getReadAheadAllowed()

### Valores

#### AS\_DEST

Determine se a leitura antecipada é permitida consultando a definição de fila ou tópico. Esse é o valor padrão nas ferramentas administrativas.

Use WMQConstants.WMQ\_READ\_AHEAD\_ALLOWED\_AS\_DEST em programas..

#### AS\_Q\_DEF

Determine se a leitura antecipada é permitida referindo-se à definição de fila

Use WMQConstants.WMQ\_READ\_AHEAD\_ALLOWED\_AS\_Q\_DEF em programas..

#### AS\_TOPIC\_DEF

Determine se a leitura antecipada é permitida referindo-se à definição de tópico

Use WMQConstants.WMQ\_READ\_AHEAD\_ALLOWED\_AS\_TOPIC\_DEF em programas..

#### NÃO

A leitura antecipada não é permitida

Use o WMQConstants.WMQ\_READ\_AHEAD\_ALLOWED\_DISABLED em programas

#### SIM

A leitura antecipada é permitida

Use WMQConstants.WMQ\_READ\_AHEAD\_ALLOWED\_ENABLED em programas..

## READAHEADCLOSEPOLICY

Para mensagens que estão sendo entregues para um listener de mensagem assíncrona, o que acontece com as mensagens no buffer de leitura antecipada interno quando o consumidor de mensagem é fechado

### Objetos Aplicáveis

Fila, Tópico

Nome longo da ferramenta de administração do JMS : READAHEADCLOSEPOLICY

Nome abreviado da ferramenta de administração do JMS : RACP

## Acesso programático

Setters / getters

- `MQDestination.setReadAheadClosePolicy()`
- `MQDestination.getReadAheadClosePolicy()`

## Valores

### **DELIVER\_ALL**

Todas as mensagens no buffer de leitura antecipada interno são entregues para o listener de mensagem do aplicativo antes de retornar. Esse é o valor padrão nas ferramentas administrativas.

Use `WMQConstants.WMQ_READ_AHEAD_DELIVERALL` em programas..

### **DELIVER\_CURRENT**

Apenas a chamada do listener de mensagem atual é concluída antes de retornar, potencialmente deixando mensagens no buffer de leitura antecipada interno, que são, então, descartados

Use `WMQConstants.WMQ_READ_AHEAD_DELIVERCURRENT` em programas..

## RECEIVECCSID

A propriedade de destino que configura o destino CCSID para a conversão de mensagens do gerenciador de filas. O valor é ignorado, a menos que `RECEIVECONVERSION` seja configurado como `WMQ_RECEIVE_CONVERSION_QMGR`

## Objetos Aplicáveis

Fila, Tópico

Nome longo da ferramenta de administração do JMS : `RECEIVECCSID`

Nome abreviado da ferramenta de administração do JMS : `RCCS`.

## Acesso programático

### **Configuradores / Getters**

- `MQDestination.setReceiveCCSID`
- `MQDestination.getReceiveCCSID`

## Valores

### **WMQConstants.WMQ\_RECEIVE\_CC\_SID\_JVM\_DEFAULT**

**0** - Usar `JVM Charset.defaultCharset`

### **1208**

UTF-8

### **ccsid**

Identificador do conjunto de caracteres codificados suportado

## RECEIVECONVERSION

A propriedade de destino que determina se a conversão de dados será executada pelo gerenciador de filas

## Objetos Aplicáveis

Fila, Tópico

Nome longo da ferramenta de administração do JMS : `RECEIVECONVERSION`

Nome abreviado da ferramenta de administração do JMS : RCNV.

## Acesso programático

### Configuradores / Getters

- `MQDestination.setReceiveConversion`
- `MQDestination.getReceiveConversion`

### Valores

#### **WMQConstants.WMQ\_RECEIVE\_CONVERSION\_CLIENT\_MSG**

1 -Executar somente a conversão de dados no cliente JMS O valor padrão de até V7.0e de, e incluindo, 7.0.1.5.

#### **WMQConstants.WMQ\_RECEIVE\_CONVERSION\_QMGR**

2 -Executar conversão de dados no gerenciador de filas antes de enviar uma mensagem ao cliente. O valor padrão (e único) de V7.0 para V7.0.1.4 inclusive, exceto se o APAR IC72897 for aplicado.

## RECEIVEISOLATION

Essa propriedade determina se um assinante pode receber mensagens não confirmadas na fila de assinantes.

### Objetos Aplicáveis

ConnectionFactory, TopicConnectionFactory

Nome longo da ferramenta de administração do JMS : RECEIVEISOLATION

Nome abreviado da ferramenta de administração do JMS : RCVISOL

### Valores

#### **CONFIRMADO**

Um assinante recebe apenas as mensagens na fila de assinantes que foram confirmados Esse é o valor padrão nas ferramentas administrativas.

Use `WMQConstants.WMQ_RCVISOL_COMMITTED` em programas..

#### **NÃO CONFIRMADO**

Um assinante pode receber mensagens que não foram consolidadas na fila de assinantes.

Use `WMQConstants.WMQ_RCVISOL_UNCOMMITTED` em programas..

## RECEXIT

Identifica uma saída de recebimento de canal, ou uma sequência de saídas de recebimento, a ser executada sucessivamente

Configuração adicional pode ser necessária para que o IBM MQ classes for JMS localize saídas de recebimento. Para obter mais informações, consulte [Configurando as classes IBM MQ para JMS para usar saídas de canal.](#)

### Objetos Aplicáveis

ConnectionFactory, QueueConnectionFactory, TopicConnectionFactory, XAConnectionFactory, XAQueueConnectionFactory, XATopicConnectionFactory.

Nome longo da ferramenta de administração do JMS : RECEXIT

Nome abreviado da ferramenta de administração do JMS : RCX.

## Acesso programático

Setters / getters

- MQConnectionFactory.setReceiveExit ()
- MQConnectionFactory.getReceiveExit ()

## Valores

- `Null`. Esse é o valor-padrão.
- Uma sequência composta por um ou mais itens separados por vírgulas, em que cada item é:
  - O nome de uma classe que implementa a interface `WMQReceiveExit` (para uma saída de recebimento de canal escrita em Java).
  - Uma sequência no formato `libraryName(entryPointName)` (para uma saída de recebimento do canal não gravada em Java)

## RECEXITINIT

Os dados do usuário que são passados para as saídas de recebimento do canal quando são chamados.

### Objetos Aplicáveis

ConnectionFactory, QueueConnectionFactory, TopicConnectionFactory, XAConnectionFactory, XAQueueConnectionFactory, XATopicConnectionFactory.

Nome longo da ferramenta de administração do JMS : RECEXITINIT

Nome abreviado da ferramenta de administração do JMS : RCXI

## Acesso programático

Setters / getters

- MQConnectionFactory.setReceiveExitInit()
- MQConnectionFactory.getReceiveExitInit()

## Valores

nulo

Uma sequência que inclui um ou mais itens de dados do usuário separados por vírgulas. Esse é o valor-padrão.

## REPLYTOSTYLE

Determina como o campo `JMSReplyTo` em uma mensagem recebida é construído

### Objetos Aplicáveis

ConnectionFactory, QueueConnectionFactory, TopicConnectionFactory, XAConnectionFactory, XAQueueConnectionFactory, XATopicConnectionFactory.

Nome longo da ferramenta de administração do JMS : REPLYTOSTYLE

Nome abreviado da ferramenta de administração do JMS : RTOST

## Acesso programático

Setters / getters

- MQConnectionFactory.setReplyToStyle()

- `MQConnectionFactory.getReplyToStyle()`

## Valores

### DEFAULT

Equivalente a MQMD

### RFH2

Use o valor fornecido no cabeçalho RFH2 . Se um valor `JMSReplyTo` foi configurado no aplicativo de envio, use esse valor.

### MQMD

Use o valor fornecido pelo MQMD Esse comportamento é equivalente ao comportamento padrão do IBM WebSphere MQ 6.0.2.4 e do 6.0.2.5

Se o valor `JMSReplyTo` configurado pelo aplicativo de envio não contiver um nome do gerenciador de filas, o gerenciador de filas de recebimento inserirá seu próprio nome no MQMD.. Se você configurar esse parâmetro para MQMD, a fila de resposta usada estará no gerenciador de filas de recebimento. Se você configurar esse parâmetro como RFH2, a fila de resposta que você usa estará no gerenciador de filas especificado no RFH2 da mensagem enviada como originalmente configurado pelo aplicativo de envio.

Se o valor `JMSReplyTo` configurado pelo aplicativo de envio contiver um nome do gerenciador de filas, o valor desse parâmetro não será importante porque o MQMD e RFH2 contêm o mesmo valor.

## RESCANINT

Quando um consumidor de mensagens no domínio ponto a ponto usa um seletor de mensagens para selecionar quais mensagens deseja receber, IBM MQ classes for JMS procure na fila IBM MQ mensagens adequadas na sequência determinada pelo atributo `MsgDeliverySequence` da fila.

Após IBM MQ classes for JMS localizar uma mensagem adequada e entregá-la ao consumidor, IBM MQ classes for JMS continue a procura pela próxima mensagem adequada a partir de sua posição atual na fila. IBM MQ classes for JMS continuar a procurar a fila dessa maneira até atingir o final da fila ou até o intervalo de tempo em milissegundos, conforme determinado pelo valor dessa propriedade, ter expirado. Em cada caso, IBM MQ classes for JMS retorna ao início da fila para continuar a procura e um novo intervalo de tempo começa.

## Objetos Aplicáveis

`ConnectionFactory`, `QueueConnectionFactory`, `XAConnectionFactory`, `XAQueueConnectionFactory`

Nome longo da ferramenta de administração do JMS : RESCANINT

Nome abreviado da ferramenta de administração do JMS : RINT.

## Acesso programático

Setters / getters

- `MQConnectionFactory.setRescanInterval ()`
- `MQConnectionFactory.getRescanInterval ()`

## Valores

### 5000

Qualquer número inteiro positivo pode ser o valor padrão.

## SECEXIT

Identifica uma saída de segurança do canal.



Configuração adicional pode ser necessária para que o IBM MQ classes for JMS localize saídas de segurança. Para obter mais informações, consulte [Configurando as classes IBM MQ para JMS para usar saídas de canal](#).

## Objetos Aplicáveis

ConnectionFactory, QueueConnectionFactory, TopicConnectionFactory, XAConnectionFactory, XAQueueConnectionFactory, XATopicConnectionFactory.

Nome longo da ferramenta de administração do JMS : SECEXIT

Nome abreviado da ferramenta de administração do JMS : SXC

## Acesso programático

Setters / getters

- MQConnectionFactory.setSecurityExit ()
- MQConnectionFactory.getSecurityExit ()

## Valores

- `Null`. Esse é o valor-padrão.
- Uma sequência composta por um ou mais itens separados por vírgulas, em que cada item é:
  - O nome de uma classe que implementa a interface `WMQSecurityExit` (para uma saída de segurança do canal gravada em Java).
  - Uma sequência no formato `libraryName(entryPointName)` (para uma saída de segurança do canal não gravada em Java)..

## SECEXITINIT

Os dados do usuário que são transmitidos para uma saída de segurança do canal quando ela é chamada.

## Objetos Aplicáveis

ConnectionFactory, QueueConnectionFactory, TopicConnectionFactory, XAConnectionFactory, XAQueueConnectionFactory, XATopicConnectionFactory.

Nome longo da ferramenta de administração do JMS : SECEXITINIT

Nome abreviado da ferramenta de administração do JMS : SCXI

## Acesso programático

Setters / getters

- MQConnectionFactory.setSecurityExitInit()
- MQConnectionFactory.getSecurityExitInit()

## Valores

nulo

Qualquer sequência pode ser o valor padrão.

## SENDCHECKCOUNT

O número de chamadas de envio a serem permitidas entre a verificação de erros de colocação assíncronos em uma única sessão do JMS não transicionada.

## Objetos Aplicáveis

ConnectionFactory, QueueConnectionFactory, TopicConnectionFactory, XAConnectionFactory, XAQueueConnectionFactory, XATopicConnectionFactory.

Nome longo da ferramenta de administração do JMS : SENDCHECKCOUNT

Nome abreviado da ferramenta de administração do JMS : SCC

## Acesso programático

Setters / getters

- MQConnectionFactory.setSendCheckCount()
- MQConnectionFactory.getSendCheckCount()

## Valores

nulo

Qualquer sequência pode ser o valor padrão.

## SENDEXIT

Identifica uma saída de envio de canal ou uma sequência de saídas de envio a serem executadas em sucessão.

Configuração adicional pode ser necessária para que o IBM MQ classes for JMS localize saídas de envio. Para obter mais informações, consulte [Configurando as classes IBM MQ para JMS para usar saídas de canal](#).

## Objetos Aplicáveis

ConnectionFactory, QueueConnectionFactory, TopicConnectionFactory, XAConnectionFactory, XAQueueConnectionFactory, XATopicConnectionFactory.

Nome longo da ferramenta de administração do JMS : SENDEXIT

Nome abreviado da ferramenta de administração do JMS : SDX

## Acesso programático

Setters / getters

- MQConnectionFactory.setSendExit ()
- MQConnectionFactory.getSendExit ()

## Valores

- `Null`. Esse é o valor-padrão.
- Uma sequência composta por um ou mais itens separados por vírgulas, em que cada item é:
  - O nome de uma classe que implementa a interface `WMQSendExit` (para uma saída de envio de canal gravada em Java)
  - Uma sequência no formato `libraryName(entryPointName)` (para uma saída de envio do canal não gravada em Java)

## SENDEXITINIT

Os dados do usuário que são transmitidos para as saídas de envio do canal quando são chamadas.

## Objetos Aplicáveis

ConnectionFactory, QueueConnectionFactory, TopicConnectionFactory, XAConnectionFactory, XAQueueConnectionFactory, XATopicConnectionFactory.

Nome longo da ferramenta de administração do JMS : SENDEXITINIT

Nome abreviado da ferramenta de administração do JMS : SDXI

## Acesso programático

Setters / getters

- MQConnectionFactory.setSendExitInit()
- MQConnectionFactory.getSendExitInit()

## Valores

### nulo

Qualquer sequência composta por um ou mais itens de dados do usuário separados por vírgulas pode ser o valor padrão.

## SHARECONVALLOWED

Para aplicativos que usam o modo normal do provedor de sistemas de mensagens do IBM MQ ou o modo normal com restrições, essa propriedade determina se a função de conversas de compartilhamento é usada para conexões, sessões e contextos do JMS criados a partir do connection factory.

## Objetos Aplicáveis

ConnectionFactory, QueueConnectionFactory, TopicConnectionFactory, XAConnectionFactory, XAQueueConnectionFactory, XATopicConnectionFactory.

Nome longo da ferramenta de administração do JMS : SHARECONVALLOWED

Nome abreviado da ferramenta de administração do JMS : SCALD

## Acesso programático

Setters / getters

- MQConnectionFactory.setShareConvAllowed()
- MQConnectionFactory.getShareConvAllowed()

## Valores

### SIM

Conexões, sessões e contextos do JMS que são criados a partir do connection factory dentro da mesma JVM podem compartilhar uma instância de canal (que mapeia para uma conexão TCP/IP) onde apropriado.

Este é o valor padrão para ferramentas administrativas.

Para programas, use WMQConstants.WMQ\_SHARE\_CONV\_ALLOWED\_YES..

### NÃO

Cada conexão do JMS criada a partir do connection factory e cada sessão do JMS criada a partir dessas conexões do JMS, tem sua própria instância do canal (conexão TCP/IP) para um gerenciador de filas.

Para contextos JMS , o primeiro contexto criado a partir da connection factory cria duas instâncias de canal (conexões TCP/IP). Outros contextos JMS que são criados a partir do primeiro têm sua própria instância de canal (conexão TCP/IP).

Para programas, use `WMQConstants.WMQ_SHARE_CONV_ALLOWED_NO`.

### **Conceitos relacionados**

[Modos de operação do provedor de sistemas de mensagens do IBM MQ](#)  
[Compartilhando uma conexão TCP/IP nas classes do IBM MQ para JMS](#)

## **SPARSESUBS**

Controla a política de recuperação de mensagens de um objeto `TopicSubscriber`.

### **Objetos Aplicáveis**

`ConnectionFactory`, `TopicConnectionFactory`

Nome longo da ferramenta de administração do JMS : `SPARSESUBS`

Nome abreviado da ferramenta de administração do JMS : `SSUBS`

### **Acesso programático**

Setters / getters

- `MQConnectionFactory.setSparseAssinaturas ()`
- `MQConnectionFactory.getSparseAssinaturas ()`

### **Valores**

#### **NÃO**

As assinaturas recebem mensagens correspondentes frequentes. Este é o valor padrão para ferramentas administrativas.

Para programas, utilize `false`.

#### **SIM**

As assinaturas recebem mensagens de correspondência pouco frequentes. Esse valor requer que a fila de assinaturas possa ser aberta para procurar.

Para programas, utilize `true`.

## **SSLCIPHERSUITE**

O `CipherSuite` a ser usado para uma conexão TLS.

### **Objetos Aplicáveis**

`ConnectionFactory`, `QueueConnectionFactory`, `TopicConnectionFactory`, `XAConnectionFactory`, `XAQueueConnectionFactory`, `XATopicConnectionFactory`.

Nome longo da ferramenta de administração do JMS : `SSLCIPHERSUITE`

Nome abreviado da ferramenta de administração do JMS : `SCPHS`.

### **Acesso programático**

Setters / getters

- `MQConnectionFactory.setSSLCipherSuite ()`
- `MQConnectionFactory.getSSLCipherSuite ()`

### **Valores**

#### **nulo**

Esse é o valor-padrão. Para obter mais informações, consulte [Propriedades TLS de JMS objetos](#)..

## SSLCRL

Servidores de CRL para verificar a revogação de certificado TLS

### Objetos Aplicáveis

ConnectionFactory, QueueConnectionFactory, TopicConnectionFactory, XAConnectionFactory, XAQueueConnectionFactory, XATopicConnectionFactory.

Nome longo da ferramenta de administração do JMS : SSLCRL

Nome abreviado da ferramenta de administração do JMS : SCRL

### Acesso programático

Setters / getters

- MQConnectionFactory.setSSLCertLojas ()
- MQConnectionFactory.getSSLCertLojas ()

### Valores

#### nulo

Lista separada por espaço de URLs do LDAP Esse é o valor-padrão. Para obter mais informações, consulte [Propriedades TLS de JMS objetos](#)..

## SSLFIPSREQUIRED

Essa propriedade determina se uma conexão TLS deve usar um CipherSuite que seja suportado pelo IBM Java provedor JSSE FIPS (IBMJSSEFIPS).

**Nota:** No AIX, Linux, and Windows, IBM MQ fornece conformidade FIPS 140-2 por meio do módulo criptográfico IBM Crypto for C (ICC) . O certificado deste módulo foi movido para o status Histórico. Os clientes devem visualizar o [IBM Crypto for C \(ICC\) certificado](#) e estar ciente de qualquer aviso fornecido pelo NIST Um módulo FIPS 140-3 de substituição está atualmente em andamento e seu status pode ser visualizado procurando por ele na [NIST CMVP modules in process list](#).

### Objetos Aplicáveis

ConnectionFactory, QueueConnectionFactory, TopicConnectionFactory, XAConnectionFactory, XAQueueConnectionFactory, XATopicConnectionFactory.

Nome longo da ferramenta de administração do JMS : SSLFIPSREQUIRED

Nome abreviado da ferramenta de administração do JMS : SFIPS

### Acesso programático

Setters / getters

- MQConnectionFactory.setSSLFipsRequired ()
- MQConnectionFactory.getSSLFipsNecessário ()

### Valores

#### NÃO

Uma conexão TLS pode usar qualquer CipherSuite que não seja suportado pelo provedor JSSE FIPS IBM Java (IBMJSSEFIPS).

Esse é o valor-padrão. Em programas, use false

#### SIM

Uma conexão TLS deve usar um CipherSuite suportado pelo IBMJSSEFIPS.

Em programas, use true.

## SSLPEERNAME

Para TLS, uma estrutura de *nome distinto* que deve corresponder àquela fornecida pelo gerenciador de fila.

### Objetos Aplicáveis

ConnectionFactory, QueueConnectionFactory, TopicConnectionFactory, XAConnectionFactory, XAQueueConnectionFactory, XATopicConnectionFactory.

Nome longo da ferramenta de administração do JMS : SSLPEERNAME

Nome abreviado da ferramenta de administração do JMS : SPEER.

### Acesso programático

Setters / getters

- MQConnectionFactory.setSSLPeerNome ()
- MQConnectionFactory.getSSLPeerNome ().

### Valores

nulo

Esse é o valor-padrão. Para obter mais informações, consulte [Propriedades TLS de JMS objetos..](#)

## SSLRESETCOUNT

Para TLS, o número total de bytes enviados e recebidos por uma conexão antes que a chave secreta usada para criptografia seja renegociada.

### Objetos Aplicáveis

ConnectionFactory, QueueConnectionFactory, TopicConnectionFactory, XAConnectionFactory, XAQueueConnectionFactory, XATopicConnectionFactory.

Nome longo da ferramenta de administração do JMS : SSLRESETCOUNT.

Nome abreviado da ferramenta de administração do JMS : SRC

### Acesso programático

Setters / getters

- MQConnectionFactory.setSSLResetContagem ()
- MQConnectionFactory.getSSLResetContagem ()

### Valores

0

Zero ou qualquer número inteiro positivo menor ou igual a 999, 999, 999. Esse é o valor-padrão. Para obter mais informações, consulte [Propriedades TLS de JMS objetos..](#)

## STATREFRESHINT

O intervalo, em milissegundos, entre atualizações da transação de execução longa, que detecta quando um assinante perde sua conexão com o gerenciador de filas.

Esta propriedade é relevante apenas se SUBSTORE possuir o valor QUEUE.

## Objetos Aplicáveis

ConnectionFactory, TopicConnectionFactory, XAConnectionFactory, XATopicConnectionFactory

Nome longo da ferramenta de administração do JMS : STATREFRESHINT

Nome abreviado da ferramenta de administração do JMS : SRI

## Acesso programático

Setters / getters

- MQConnectionFactory.setStatusRefreshInterval()
- MQConnectionFactory.getStatusRefreshInterval()

## Valores

### 60000

Qualquer número inteiro positivo pode ser o valor padrão. Para obter mais informações, consulte [Propriedades TLS de JMS objetos..](#)

## SUBSTORE

Onde IBM MQ classes for JMS armazena dados persistentes relacionados a assinaturas ativas.

## Objetos Aplicáveis

ConnectionFactory, TopicConnectionFactory, XAConnectionFactory, XATopicConnectionFactory

Nome longo da ferramenta de administração do JMS : SUBSTORE

Nome abreviado da ferramenta de administração JMS : SS

## Acesso programático

Setters / getters

- MQConnectionFactory.setSubscriptionStore ()
- MQConnectionFactory.getSubscriptionStore ()

## Valores

### BROKER

Use o armazenamento de assinatura baseado em broker para manter detalhes de assinaturas. Este é o valor padrão para ferramentas administrativas.

Para programas, use WMQConstants.WMQ\_SUBSTORE\_BROKER..

### MIGRATE

Transfira informações de assinatura do armazenamento de assinatura baseado em fila para o armazenamento de assinatura baseado no broker

Para programas, use WMQConstants.WMQ\_SUBSTORE\_MIGRATE..

### FILA

Use o armazenamento de assinatura baseado em fila para conter detalhes das assinaturas

Para programas, use WMQConstants.WMQ\_SUBSTORE\_QUEUE

## SYNCPPOINTALLGETS

Essa propriedade determina se todas as obtenções devem ser executadas sob o ponto de sincronização

## Objetos Aplicáveis

ConnectionFactory, QueueConnectionFactory, TopicConnectionFactory, XAConnectionFactory, XAQueueConnectionFactory, XATopicConnectionFactory.

Nome longo da ferramenta de administração do JMS : SYNCPOINTALLGETS

Nome abreviado da ferramenta de administração do JMS : SPAG

## Acesso programático

Setters / getters

- MQConnectionFactory.setSyncpointAllGets()
- MQConnectionFactory.getSyncpointAllGets()

## Valores

### No

Esse é o valor-padrão.

### Sim

## TARGCLIENT

Essa propriedade determina se o formato IBM MQ RFH2 é usado para trocar informações com aplicativos de destino.

## Objetos Aplicáveis

Fila, Tópico

Nome longo da ferramenta de administração do JMS : TARGCLIENT

Nome abreviado da ferramenta de administração do JMS : TC

## Acesso programático

Setters / getters

- MQDestination.setTargetClient()
- MQDestination.getTargetClient()

## Valores

### JMS

O destino da mensagem é um aplicativo JMS .. Este é o valor padrão para ferramentas administrativas.

Para programas, use WMQConstants.WMQ\_CLIENT\_JMS\_COMPLIANT..

### MQ

O destino da mensagem é um aplicativo nãoJMS IBM MQ .

Para programas, use WMQConstants.WMQ\_CLIENT\_NONJMS\_MQ..

## TARGCLIENTMATCHING

Essa propriedade determina se uma mensagem de resposta, enviada para a fila identificada pelo campo de cabeçalho JMSReplyTo de uma mensagem recebida, possui um cabeçalho MQRFH2 somente se a mensagem recebida tiver um cabeçalho MQRFH2 .



## Objetos Aplicáveis

ConnectionFactory, QueueConnectionFactory, XAConnectionFactory, XAQueueConnectionFactory

Nome longo da ferramenta de administração do JMS : TARGCLIENTMATCHING

Nome abreviado da ferramenta de administração do JMS : TCM

## Acesso programático

Setters / getters

- MQConnectionFactory.setTargetClientMatching()
- MQConnectionFactory.getTargetClientMatching()

## Valores

### SIM

Se uma mensagem recebida não tiver um cabeçalho MQRFH2 , a propriedade TARGCLIENT do objeto Queue derivado do campo de cabeçalho JMSReplyTo da mensagem será enviada para MQ. Se a mensagem tiver um cabeçalho MQRFH2 , a propriedade TARGCLIENT será configurada como JMS no lugar. Este é o valor padrão para ferramentas administrativas.

Para programas, utilize true.

### NÃO

A propriedade TARGCLIENT do objeto Queue derivada do campo de cabeçalho JMSReplyTo de uma mensagem recebida é sempre configurada como JMS

Para programas, utilize false.

## TEMPMODEL

O nome da fila modelo a partir da qual as filas temporárias do JMS são criadas.

## Objetos Aplicáveis

ConnectionFactory, QueueConnectionFactory, XAConnectionFactory, XAQueueConnectionFactory

Nome longo da ferramenta de administração do JMS : TEMPMODEL

Nome abreviado da ferramenta de administração do JMS : TM.

## Acesso programático

Setters / getters

- MQConnectionFactory.setTemporaryModel ()
- MQConnectionFactory.getTemporaryModelo ()

## Valores

### SYSTEM.DEFAULT.MODEL.QUEUE

Qualquer sequência pode ser o valor padrão.

## TEMPQPREFIX

O prefixo que é usado para formar o nome de uma fila dinâmica do IBM MQ.

## Objetos Aplicáveis

ConnectionFactory, QueueConnectionFactory, XAConnectionFactory, XAQueueConnectionFactory

Nome longo da ferramenta de administração do JMS : TEMPQPREFIX

Nome abreviado da ferramenta de administração do JMS : TQP

## Acesso programático

Setters / getters

- MQConnectionFactory.setTempQPrefix ()
- MQConnectionFactory.getTempQPrefix ()

## Valores

### "" (sequência vazia)

O prefixo usado é CSQ.\* on z/OS e AMQ.\* em todas as outras plataformas Esses são os valores padrão

### *prefixo de fila*

O prefixo da fila é qualquer sequência que esteja em conformidade com as regras para formar conteúdo do campo *DynamicQName* em um descritor de objeto IBM MQ (estrutura MQOD), mas o último caractere não em branco deve ser um asterisco.

## TEMPTOPICPREFIX

Ao criar tópicos temporários, o JMS gera uma cadeia de tópicos no formato " TEMP /*TEMPTOPICPREFIX*/*unique\_id* " ou se essa propriedade for deixada com o valor padrão, apenas " TEMP /*unique\_id* ". Especificar um TEMPTOPICPREFIX não vazio permite que filas modelo específicas sejam definidas para criar as filas gerenciadas para assinantes para tópicos temporários criados nessa conexão.

## Objetos Aplicáveis

ConnectionFactory, TopicConnectionFactory, XAConnectionFactory, XATopicConnectionFactory

Nome longo da ferramenta de administração do JMS : TEMPTOPICPREFIX

Nome abreviado da ferramenta de administração do JMS : TTP

## Acesso programático

Setters / getters

- MQConnectionFactory.setTempTopicPrefix()
- MQConnectionFactory.getTempTopicPrefix()

## Valores

Qualquer sequência não nula que consista apenas em caracteres válidos para uma sequência de tópicos IBM MQ O valor padrão é "" (sequência vazia).

## tópico

O nome do destino do tópico JMS , este valor é usado pelo gerenciador de fila como a sequência de tópicos de uma publicação ou assinatura..

## Objetos Aplicáveis

Tópico

Nome longo da ferramenta de administração do JMS : TOPIC

Nome abreviado da ferramenta de administração do JMS : TOP

## Valores

### Qualquer sequência

Uma sequência que forma uma sequência de tópicos IBM MQ válida. Ao usar IBM MQ como um provedor de sistemas de mensagens com WebSphere Application Server, especifique um valor que corresponda ao nome pelo qual o tópico é conhecido para propósitos administrativos no WebSphere Application Server.

### Referências relacionadas

[Sequências de tópicos](#)

## TRANSPORT

A natureza de uma conexão com um gerenciador de filas ou um broker

### Objetos Aplicáveis

ConnectionFactory, QueueConnectionFactory, TopicConnectionFactory, XAConnectionFactory, XAQueueConnectionFactory, XATopicConnectionFactory.

Nome longo da ferramenta de administração do JMS : TRANSPORT

Nome abreviado da ferramenta de administração JMS : TRAN

### Acesso programático

Setters / getters

- MQConnectionFactory.setTransportTipo ()
- MQConnectionFactory.getTransportTipo ()

## Valores

### BIND

Para obter uma conexão com um gerenciador de fila no modo de ligações Este é o valor padrão para ferramentas administrativas.

Para programas, use WMQConstants.WMQ\_CM\_BINDINGS..

### CLIENTE

Para uma conexão com um gerenciador de filas no modo de cliente

Para programas, use WMQConstants.WMQ\_CM\_CLIENT..

### Direto

Para uma conexão em tempo real com um broker que não usa o túnel HTTP .

Para programas, use WMQConstants.WMQ\_CM\_DIRECT\_TCPIP..

### DIRECTHTTP

Para uma conexão em tempo real com um broker usando o túnel HTTP . Apenas HTTP 1.0 é suportado

Para programas, use WMQConstants.WMQ\_CM\_DIRECT\_HTTP..

### Conceitos relacionados

[“Dependências entre as propriedades de objetos do IBM MQ classes for JMS”](#) na página 1964

A validade de algumas propriedades é dependente dos valores específicos de outras propriedades

## WILDCARDFORMAT

Essa propriedade determina qual versão da sintaxe curinga será usada.

## Objetos Aplicáveis

ConnectionFactory, TopicConnectionFactory, XAConnectionFactory, XATopicConnectionFactory

Nome longo da ferramenta de administração do JMS : WILDCARDFORMAT

Nome abreviado da ferramenta de administração do JMS : WCFMT

## Acesso programático

Setters / getters

- MQConnectionFactory.setWildcardFormat()
- MQConnectionFactory.getWildcardFormat()

## Valores

### TOPIC\_ONLY

Reconhece apenas curingas de nível de tópico, conforme usado no broker versão 2. Este é o valor padrão para ferramentas administrativas.

Para programas, use WMQConstants.WMQ\_WILDCARD\_TOPIC\_ONLY..

### CHAR\_ONLY

Reconhece caracteres curingas apenas, conforme usado no broker versão 1.

Para programas, use WMQConstants.WMQ\_WILDCARD\_CHAR\_ONLY..

## A propriedade ENCODING

A propriedade ENCODING compreende três subpropriedades, em doze combinações possíveis.

Os valores válidos que a propriedade ENCODING pode tomar são construídos a partir das três subpropriedades:

### Codificação de Inteiro

Normal ou invertido

### Codificação Decimal

Normal ou invertido

### codificação de vírgula flutuante

IEEE normal, IEEE revertido ou z/OS

A propriedade ENCODING é expressa como uma sequência de três caracteres com a sintaxe a seguir:

```
{N|R}{N|R}{N|R|3}
```

Nesta cadeia:

- N denota normal
- R denota reverso
- 3 denota z/OS
- O primeiro caractere representa *codificação de número inteiro*
- O segundo caractere representa *codificação decimal*
- O terceiro caractere representa a *codificação de vírgula flutuante*

Isso fornece um conjunto de doze valores possíveis para a propriedade ENCODING

Há um valor adicional, a sequência NATIVE, que configura valores de codificação apropriados para a plataforma Java .

Os exemplos a seguir mostram combinações válidas para ENCODING:

```
ENCODING (NMR)
ENCODING (NATIVE)
ENCODING (RR3)
```

## Propriedades de TLS de objetos do JMS

Ative a criptografia de Segurança da Camada de Transporte (TLS) usando a propriedade SSLCIPHERSUITE. É possível então mudar as características da criptografia TLS usando várias outras propriedades.

Ao especificar TRANSPORT (CLIENT), é possível ativar a comunicação criptografada TLS usando a propriedade SSLCIPHERSUITE. Configure essa propriedade como um CipherSuite válido fornecido por seu provedor JSSE; ele deve corresponder ao CipherSpec nomeado no canal SVRCONN nomeado pela propriedade CHANNEL..

No entanto, CipherSpecs (conforme especificado no canal SVRCONN) e CipherSuites (conforme especificado nos objetos ConnectionFactory ) usam diferentes esquemas de nomenclatura para representar os mesmos algoritmos de criptografia TLS. Se um nome CipherSpec reconhecido for especificado na propriedade SSLCIPHERSUITE, JMSAdmin emitirá um aviso e mapeará o CipherSpec para seu CipherSuiteequivalente. Consulte [TLS CipherSpecs e CipherSuites no IBM MQ classes for JMS](#) para obter uma lista de CipherSpecs reconhecidos pelo IBM MQ e JMSAdmin.

Se precisar de uma conexão para usar um CipherSuite que seja suportado pelo IBM Java provedor JSSE FIPS (IBMJSSEFIPS), configure a propriedade SSLFIPSREQUIRED do connection factory como YES. O valor padrão dessa propriedade é NO, que significa que uma conexão pode usar qualquer CipherSuitesuportado. A propriedade será ignorada se SSLCIPHERSUITE não estiver configurado

O SSLPEERNAME corresponde ao formato do parâmetro SSLPEER, que pode ser configurado nas definições de canal.. É uma lista de pares nome-valor de atributo separados por vírgulas ou pontos e vírgulas. Por exemplo:

```
SSLPEERNAME (CN=QMGR.*, OU=IBM, OU=WEBSHERE)
```

O conjunto de nomes e valores compõe um *nome distinto*. Para obter mais detalhes sobre nomes distintos e seu uso com IBM MQ, consulte [Protegendo IBM MQ](#).

O exemplo fornecido verifica o certificado de identificação apresentado pelo servidor no momento da conexão.. Para que a conexão seja bem-sucedida, o certificado deve ter um Nome Comum iniciando QMGR., e deve ter pelo menos dois nomes de Unidade Organizacional, o primeiro dos quais é IBM e o segundo WEBSHERE. A verificação não faz distinção entre maiúsculas e minúsculas

Se SSLPEERNAME não estiver configurado, nenhuma verificação será executada. SSLPEERNAME será ignorado se SSLCIPHERSUITE não estiver configurado

A propriedade SSLCRL especifica zero ou mais servidores CRL (Certificate Revocation List). O uso dessa propriedade requer uma JVM em Java 2 v1.4.. Esta é uma lista delimitada por espaços de entrada do formulário:

```
ldap:// hostname:[ port ]
```

opcionalmente seguido por um único /. Se *port* for omitido, a porta LDAP padrão de 389 será assumida. No momento da conexão, o certificado TLS apresentado pelo servidor é verificado com relação aos servidores de CRL especificados. Consulte [Protegendo IBM MQ](#) para obter mais sobre a segurança CRL.

Se SSLCRL não for configurado, nenhuma verificação será executada. SSLCRL será ignorado se SSLCIPHERSUITE não estiver configurado

A propriedade SSLRESETCOUNT representa o número total de bytes enviados e recebidos por uma conexão antes que a chave secreta usada para criptografia seja renegociada. O número de bytes enviados é o número antes da criptografia e o número de bytes recebidos é o número após a decriptografia. O

número de bytes também inclui informações de controle enviadas e recebidas pelo IBM MQ classes for JMS.

Por exemplo, para configurar um objeto `ConnectionFactory` que pode ser usado para criar uma conexão por meio de um canal MQI ativado para TLS com uma chave secreta que seja renegociada após 4 MB de dados fluírem, emita o seguinte comando para JMSAdmin:

```
ALTER CF(my.cf) SSLRESETCOUNT(4194304)
```

Se o valor de `SSLRESETCOUNT` for zero, que é o valor padrão, a chave secreta nunca será renegociada. A propriedade `SSLRESETCOUNT` será ignorada se `SSLCIPHERSUITE` não estiver configurado.

## Referência do IBM MQ Message Service Client (XMS) for .NET

Esta seção de referência fornece informações sobre as interfaces de classe IBM MQ Message Service Client (XMS) for .NET (XMS .NET) e sobre as propriedades de objeto definidas por XMS.

### .NET interfaces

Esta seção descreve as interfaces de classe do .NET e suas propriedades e métodos

A tabela a seguir resume as interfaces, que são definidas no namespace `IBM.XMS`.

Interface	Descrição
<a href="#">“IBytesMessage” na página 2020</a>	Uma mensagem de bytes é uma mensagem cujo corpo compreende um fluxo de bytes.
<a href="#">“IConnection” na página 2030</a>	Um objeto <code>Connection</code> representa a conexão ativa do aplicativo para um servidor de sistema de mensagens.
<a href="#">“IConnectionFactory” na página 2033</a>	Um aplicativo usa um <code>connection factory</code> para criar uma conexão.
<a href="#">“IConnectionMetaDados” na página 2034</a>	Um objeto de dados <code>ConnectionMeta</code> fornece informações sobre uma conexão.
<a href="#">“IDestination” na página 2035</a>	Um destino é para onde um aplicativo envia mensagens ou é uma origem da qual um aplicativo recebe mensagens, ou ambos.
<a href="#">“ExceptionListener” na página 2036</a>	Um aplicativo usa um <code>listener</code> de exceção para ser notificado assincronamente de um problema com uma conexão.
<a href="#">“Exceção IllegalState” na página 2036</a>	XMS lança essa exceção se um aplicativo chamar um método em um momento incorreto ou inadequado ou se XMS não estiver em um estado apropriado para a operação solicitada.
<a href="#">“InitialContext” na página 2037</a>	Um aplicativo usa um objeto <code>InitialContext</code> para criar objetos de definições de objeto que são recuperados de um repositório de objetos administrados.
<a href="#">“InvalidClientIDException” na página 2039</a>	XMS lança essa exceção se um aplicativo tentar configurar um identificador de cliente para uma conexão, mas o identificador de cliente não for válido ou já estiver em uso.

Tabela 871. Resumo das interfaces de classe .NET (continuação)

<b>Interface</b>	<b>Descrição</b>
<a href="#">“Exceção de InvalidDestination” na página 2039</a>	XMS lançará essa exceção se um aplicativo especificar um destino que não seja válido
<a href="#">“InvalidSelectorExceção” na página 2039</a>	XMS emitirá essa exceção se um aplicativo fornecer uma expressão de seletor de mensagem cuja sintaxe não é válida
<a href="#">“IMapMessage” na página 2040</a>	Uma mensagem de mapa é uma mensagem cujo corpo consiste em um conjunto de pares nome-valor, em que cada valor possui um tipo de dados associado.
<a href="#">“IMessage” na página 2048</a>	Um objeto de mensagem representa uma mensagem que um aplicativo envia ou recebe. IMessage é uma superclasse para as classes de mensagem, como IMapMessage.
<a href="#">“IMessageConsumer” na página 2054</a>	Um aplicativo usa um consumidor de mensagem para receber mensagens enviadas a um destino.
<a href="#">“MessageEOFException” na página 2057</a>	XMS lança essa exceção se XMS encontrar o final de um fluxo de mensagens de bytes quando um aplicativo estiver lendo o corpo de uma mensagem de bytes.
<a href="#">“MessageFormatException” na página 2057</a>	XMS lançará essa exceção se o XMS encontrar uma mensagem com um formato que não seja válido
<a href="#">“IMessageListener (delegado)” na página 2057</a>	Um aplicativo usa um listener de mensagens para receber mensagens assincronamente.
<a href="#">“MessageNotReadableException” na página 2058</a>	XMS lança essa exceção se um aplicativo tentar ler o corpo de uma mensagem que é somente gravação.
<a href="#">“MessageNotWritableException” na página 2058</a>	XMS lança essa exceção se um aplicativo tentar gravar no corpo de uma mensagem que é somente leitura.
<a href="#">“IMessageProducer” na página 2058</a>	Um aplicativo usa um produtor de mensagem para enviar mensagens para um destino.
<a href="#">“IObjectMessage” na página 2064</a>	Uma mensagem de objeto é uma mensagem cujo corpo compreende um objeto Java ou .NET serializado.
<a href="#">“IPropertyContext” na página 2065</a>	IPropertyContext é uma superclasse abstrata que contém métodos que obtêm e configuram propriedades. Estes métodos são herdados por outras classes.
<a href="#">“IQueueBrowser” na página 2073</a>	Um aplicativo usa um navegador de filas para pesquisar mensagens em uma fila sem removê-las.
<a href="#">“Solicitante” na página 2075</a>	Um aplicativo usa um solicitante para enviar uma mensagem de solicitação e, em seguida, aguardar e receber a resposta..
<a href="#">“Exceção de ResourceAllocation” na página 2076</a>	XMS lança essa exceção se XMS não puder alocar os recursos necessários por um método.

Tabela 871. Resumo das interfaces de classe .NET (continuação)

Interface	Descrição
<a href="#">“SecurityException” na página 2077</a>	XMS lança essa exceção se o identificador de usuário e a senha fornecidos para autenticar um aplicativo forem rejeitados XMS também lança essa exceção se uma verificação de autoridade falhar e impedir que um método seja concluído.
<a href="#">“ISession” na página 2077</a>	Uma sessão é um único contexto encadeado para enviar e receber mensagens.
<a href="#">“IStreamMessage” na página 2087</a>	Uma mensagem de fluxo é uma mensagem cujo corpo compreende um fluxo de valores, em que cada valor possui um tipo de dados associado.
<a href="#">“ITextMessage” na página 2096</a>	Uma mensagem de texto é uma mensagem cujo corpo compreende uma sequência.
<a href="#">“TransactionInProgressException” na página 2097</a>	XMS lançará essa exceção se um aplicativo solicitar uma operação que não seja válida porque uma transação está em andamento
<a href="#">“TransactionRolledBackException” na página 2097</a>	XMS lança esta exceção se um aplicativo chamar Session.commit() para confirmar a transação atual, mas a transação será, então, revertida.
XMSC	Para .NET, XMS nomes de propriedades e valores são definidos nessa classe como constantes públicas. Para obter detalhes adicionais, consulte <a href="#">“Propriedades de objetos XMS” na página 2100</a> .
<a href="#">“XMSException” na página 2097</a>	Se XMS detectar um erro ao processar uma chamada para um método .NET, XMS lançará uma exceção. Uma exceção é um objeto que contém informações sobre o erro.  Há diferentes tipos de exceção XMS, e um objeto XMSException é apenas um tipo de exceção. Entretanto, a classe XMSException é uma superclasse das outras classes de exceção XMS. XMS lança um objeto XMSException em situações em que nenhum dos outros tipos de exceção é apropriado.
<a href="#">“XMSFactoryFactory” na página 2098</a>	Se um aplicativo não estiver usando objetos administrados, utilize essa classe para criar connection factories, filas e tópicos..

A definição de cada método lista os códigos de exceção que o XMS pode retornar se detectar um erro ao processar uma chamada para o método. Cada código de exceções é representado por sua constante nomeada, que possui uma exceção correspondente.

## IBytesMessage

Uma mensagem de bytes é uma mensagem cujo corpo compreende um fluxo de bytes.

### Hierarquia de herança:

```

IBM.XMS.IPropertyContext
|
+---- IBM.XMS.IMessage
    
```



```
|  
+----IBM.XMS.IBytesMessage
```

## **.NET propriedades**

*BodyLength - Obter Comprimento do Corpo*

### **Interface:**

```
Int64 BodyLength  
{  
    get;  
}
```

Obtenha o comprimento do corpo da mensagem em bytes quando o corpo da mensagem for somente leitura

O valor retornado é o comprimento do corpo inteiro, independentemente de onde o cursor para ler a mensagem está atualmente posicionado.

### **Exceções:**

- XMSEException
- MessageNotReadableException

## **Métodos**

*ReadBoolean - Valor Booleano de leitura*

### **Interface:**

```
Boolean ReadBoolean();
```

Leia um valor booleano do fluxo de mensagens de bytes.

### **Parâmetros:**

Nenhum

### **Retorna:**

O valor booleano que é lido.

### **Exceções:**

- XMSEException
- MessageNotReadableException
- MessageEOFException

*ReadSignedByte-Byte de Leitura*

### **Interface:**

```
Int16 ReadSignedByte();
```

Leia o próximo byte do fluxo de mensagens de bytes como um número inteiro de 8 bits assinado.

### **Parâmetros:**

Nenhum

### **Retorna:**

O byte lido.

**Exceções:**

- XMSEException
- MessageNotReadableException
- MessageEOFException

*ReadBytes - Bytes de leitura*

**Interface:**

```
Int32 ReadBytes(Byte[] array);  
Int32 ReadBytes(Byte[] array, Int32 length);
```

Ler uma matriz de bytes a partir do fluxo de mensagens de bytes a partir da posição atual do cursor.

**Parâmetros:****matriz (saída)**

O buffer para conter a matriz de bytes que é lida. Se o número de bytes restantes a serem lidos do fluxo antes da chamada for maior ou igual ao comprimento do buffer, o buffer será preenchido. Caso contrário, o buffer será parcialmente preenchido, com todos os bytes restantes.

Se você especificar um ponteiro nulo na entrada, o método ignorará os bytes sem lê-los. Se o número de bytes restantes a serem lidos a partir do fluxo antes da chamada for maior ou igual ao comprimento do buffer, o número de bytes ignorados será igual ao comprimento do buffer. Caso contrário, todos os bytes restantes serão ignorados.. O cursor permanece na próxima posição para ser lido no fluxo de mensagens de bytes

**comprimento (entrada)**

O comprimento do buffer em bytes

**Retorna:**

O número de bytes lidos no buffer. Se o buffer estiver parcialmente preenchido, o valor será menor que o comprimento do buffer, indicando que não há mais bytes restantes para serem lidos. Se não houver bytes restantes a serem lidos do fluxo antes da chamada, o valor será XMSC\_END\_OF\_STREAM.

Se você especificar um ponteiro nulo na entrada, o método não retornará valor.

**Exceções:**

- XMSEException
- MessageNotReadableException

*ReadChar - Caractere de Leitura*

**Interface:**

```
Char ReadChar();
```

Leia os próximos 2 bytes do fluxo de mensagens de bytes como um caractere.

**Parâmetros:**

Nenhum

**Retorna:**

O caractere lido.

**Exceções:**

- XMSEException
- MessageNotReadableException
- MessageEOFException

### *ReadDouble - Ler Número de Ponto Flutuante de Precisão Dupla*

#### **Interface:**

```
Double ReadDouble();
```

Leia os próximos 8 bytes do fluxo de mensagens de bytes como um número de ponto flutuante de precisão dupla.

#### **Parâmetros:**

Nenhum

#### **Retorna:**

O número do ponto flutuante de precisão dupla que é lido

#### **Exceções:**

- XMSEException
- MessageNotReadableException
- MessageEOFException

### *ReadFloat - Ler número de ponto flutuante*

#### **Interface:**

```
Single ReadFloat();
```

Leia os próximos 4 bytes do fluxo de mensagens de bytes como um número de ponto flutuante.

#### **Parâmetros:**

Nenhum

#### **Retorna:**

O número de ponto flutuante que é lido

#### **Exceções:**

- XMSEException
- MessageNotReadableException
- MessageEOFException

### *ReadInt - Ler Número Inteiro*

#### **Interface:**

```
Int32 ReadInt();
```

Leia os próximos 4 bytes do fluxo de mensagens de bytes como um número inteiro de 32 bits assinado.

#### **Parâmetros:**

Nenhum

#### **Retorna:**

O número inteiro que é lido.

#### **Exceções:**

- XMSEException
- MessageNotReadableException
- MessageEOFException

### *ReadLong -Número inteiro longo de leitura*

#### **Interface:**

```
Int64 ReadLong();
```

Leia os próximos 8 bytes do fluxo de mensagens de bytes como um número inteiro de 64 bits assinado.

#### **Parâmetros:**

Nenhum

#### **Retorna:**

O número inteiro longo que é lido.

#### **Exceções:**

- XMSEException
- MessageNotReadableException
- MessageEOFException

### *ReadShort -Número Inteiro Curto de Leitura*

#### **Interface:**

```
Int16 ReadShort();
```

Leia os próximos 2 bytes do fluxo de mensagens de bytes como um número inteiro de 16 bits assinado.

#### **Parâmetros:**

Nenhum

#### **Retorna:**

O número inteiro curto que é lido.

#### **Exceções:**

- XMSEException
- MessageNotReadableException
- MessageEOFException

### *ReadByte -Byte não assinado de leitura*

#### **Interface:**

```
Byte ReadByte();
```

Leia o próximo byte do fluxo de mensagens de bytes como um número inteiro de 8 bits não assinado.

#### **Parâmetros:**

Nenhum

#### **Retorna:**

O byte lido.

#### **Exceções:**

- XMSEException
- MessageNotReadableException
- MessageEOFException

## *ReadUnsignedShort-Read Unsigned Short Integer*

### **Interface:**

```
Int32 ReadUnsignedShort();
```

Leia os próximos 2 bytes do fluxo de mensagens de bytes como um número inteiro de 16 bits não assinado.

### **Parâmetros:**

Nenhum

### **Retorna:**

O número inteiro curto não assinado que é lido.

### **Exceções:**

- XMSEException
- MessageNotReadableException
- MessageEOFException

## *ReadUTF -Ler Sequência UTF*

### **Interface:**

```
String ReadUTF();
```

Leia uma cadeia, codificada em UTF-8, a partir do fluxo de mensagens de bytes.

**Nota:** Antes de chamar ReadUTF(), assegure que o cursor do buffer esteja apontando para o início do fluxo de mensagens de byte.

### **Parâmetros:**

Nenhum

### **Retorna:**

Um objeto String encapsulando a sequência que é lida.

### **Exceções:**

- XMSEException
- MessageNotReadableException
- MessageEOFException

## *Reconfigurar-Reconfigurar*

### **Interface:**

```
void Reset();
```

Coloque o corpo da mensagem no modo somente leitura e reposicione o cursor no início do fluxo de mensagem de bytes..

### **Parâmetros:**

Nenhum

### **Retorna:**

Nulo

### **Exceções:**

- XMSEException
- MessageNotReadableException

*WriteBoolean -Gravar Valor Booleano*

**Interface:**

```
void WriteBoolean(Boolean value);
```

Grave um valor booleano no fluxo de mensagens de bytes.

**Parâmetros:**

**valor (entrada)**

O valor booleano a ser gravado.

**Retorna:**

Nulo

**Exceções:**

- XMSEException
- MessageNotWritableException

*WriteByte -Byte de gravação*

**Interface:**

```
void WriteByte(Byte value);  
void WriteSignedByte(Int16 value);
```

Gravar um byte no fluxo de mensagens de bytes.

**Parâmetros:**

**valor (entrada)**

O byte a ser gravado..

**Retorna:**

Nulo

**Exceções:**

- XMSEException
- MessageNotWritableException

*WriteBytes -Bytes de Gravação*

**Interface:**

```
void WriteBytes(Byte[] value);
```

Grave uma matriz de bytes no fluxo de mensagens de bytes.

**Parâmetros:**

**valor (entrada)**

A matriz de bytes a ser gravada

**Retorna:**

Nulo

**Exceções:**

- XMSEException
- MessageNotWritableException

## *WriteBytes -Matriz de bytes parciais de gravação*

### **Interface:**

```
void WriteBytes(Byte[] value, int offset, int length);
```

Grave uma matriz parcial de bytes no fluxo de mensagens de bytes, conforme definido pelo comprimento especificado.

### **Parâmetros:**

**valor (entrada)**

A matriz de bytes a ser gravada

**deslocamento (entrada)**

O ponto de início para a matriz de bytes a ser gravada

**comprimento (entrada)**

O número de bytes a serem gravados

### **Retorna:**

Nulo

### **Exceções:**

- XMSEException
- MessageNotWritableException

## *WriteChar -Caractere de gravação*

### **Interface:**

```
void WriteChar(Char value);
```

Grave um caractere no fluxo de mensagens de bytes como 2 bytes, primeiro byte de alta ordem.

### **Parâmetros:**

**valor (entrada)**

O caractere a ser gravado

### **Retorna:**

Nulo

### **Exceções:**

- XMSEException
- MessageNotWritableException

## *WriteDouble -Número do ponto flutuante de precisão dupla de gravação*

### **Interface:**

```
void WriteDouble(Double value);
```

Converta um número de vírgula flutuante de precisão dupla em um número inteiro longo e grave o número inteiro longo no fluxo de mensagens de bytes como 8 bytes, primeiro byte de alta ordem..

### **Parâmetros:**

**valor (entrada)**

O número de ponto flutuante de precisão dupla a ser gravado

### **Retorna:**

Nulo

**Exceções:**

- XMSEException
- MessageNotWritableException

*WriteFloat -Número do ponto flutuante de gravação*

**Interface:**

```
void WriteFloat(Single value);
```

Converta um número de vírgula flutuante em um número inteiro e grave o número inteiro no fluxo de mensagens de bytes como 4 bytes, primeiro byte de alta ordem.

**Parâmetros:****valor (entrada)**

O número de ponto flutuante a ser gravado

**Retorna:**

Nulo

**Exceções:**

- XMSEException
- MessageNotWritableException

*WriteInt -Número inteiro de gravação*

**Interface:**

```
void WriteInt(Int32 value);
```

Grave um número inteiro no fluxo de mensagens de bytes como 4 bytes, primeiro byte de alta ordem.

**Parâmetros:****valor (entrada)**

O número inteiro a ser gravado.

**Retorna:**

Nulo

**Exceções:**

- XMSEException
- MessageNotWritableException

*WriteLong -Número inteiro longo de gravação*

**Interface:**

```
void WriteLong(Int64 value);
```

Grave um número inteiro longo no fluxo de mensagens de bytes como 8 bytes, primeiro byte de alta ordem.

**Parâmetros:****valor (entrada)**

O número inteiro longo a ser gravado.

**Retorna:**

Nulo



**Exceções:**

- XMSEException
- MessageNotWritableException

*WriteObject -Gravar Objeto*

**Interface:**

```
void WriteObject(Object value);
```

Grave o objeto especificado no fluxo da mensagem de byte

**Parâmetros:****valor (entrada)**

O objeto a ser gravado, que deve ser uma referência a um tipo primitivo..

**Retorna:**

Nulo

**Exceções:**

- XMSEException
- MessageNotWritableException

*WriteShort -Escrever Número Inteiro Curto*

**Interface:**

```
void WriteShort(Int16 value);
```

Grave um número inteiro curto no fluxo de mensagens de bytes como 2 bytes, primeiro byte de alta ordem.

**Parâmetros:****valor (entrada)**

O número inteiro curto a ser gravado.

**Retorna:**

Nulo

**Exceções:**

- XMSEException
- MessageNotWritableException

*WriteUTF -Gravar sequência UTF*

**Interface:**

```
void WriteUTF(String value);
```

Grave uma sequência, codificada em UTF-8, para o fluxo de mensagens de bytes

**Parâmetros:****valor (entrada)**

Um objeto String encapsulando a sequência a ser gravada.

**Retorna:**

Nulo

**Exceções:**

- XMSEException

- `MessageNotWritableException`

### **Propriedades e métodos herdados**

As propriedades a seguir são herdadas da interface `IMessage`:

`JMSCorrelationID`, `JMSDeliveryMode`, `JMSDestination`, `JMSExpiration`, `JMSMessageID`, `JMSPriority`, `JMSRedelivered`, `JMSReplyTo`, `JMSTimestamp`, `JMSType`, `Propriedades`

Os métodos a seguir são herdados da interface `IMessage`:

`clearBody`, `clearProperties`, `PropertyExists`

Os métodos a seguir são herdados da interface `IPropertyContext`:

`GetBooleanProperty`, `GetByteProperty`, `GetBytesProperty`, `GetCharProperty`, `GetDoubleProperty`, `GetFloatProperty`, `GetIntProperty`, `GetLongProperty`, `GetObjectProperty`, `GetShortProperty`, `GetStringProperty`, `SetBooleanProperty`, `SetByteProperty`, `SetBytesProperty`, `SetCharProperty`, `SetDoubleProperty`, `SetFloatProperty`, `SetIntProperty`, `SetLongProperty`, `SetObjectProperty`, `SetShortProperty`, `SetStringProperty`

## **ICConnection**

Um objeto `Connection` representa a conexão ativa do aplicativo para um servidor de sistema de mensagens.

### **Hierarquia de herança:**

```
IBM.XMS.IPropertyContext
|
+---- IBM.XMS.ICConnection
```

Para obter uma lista das propriedades definidas XMS de um objeto `Connection`, consulte [“Propriedades da Conexão”](#) na página 2101.

## **.NET propriedades**

*ClientID - Obter e Configurar ID do Cliente*

### **Interface:**

```
String ClientID
{
    get;
    set;
}
```

Obter e configurar o identificador de cliente para a conexão.

O identificador de cliente pode ser pré-configurado pelo administrador em um `ConnectionFactory` designado configurando `ClientID`.

Um identificador de cliente é usado apenas para suportar assinaturas duráveis no domínio de publicação / assinatura e é ignorado no domínio ponto a ponto.

Se um aplicativo configurar um identificador de cliente para uma conexão, o aplicativo deverá fazer isso imediatamente após criar a conexão e antes de executar qualquer outra operação na conexão. Se o aplicativo tentar configurar um identificador de cliente após esse ponto, a chamada emitirá exceção `IllegalStateException`.

Essa propriedade não é válida para uma conexão em tempo real com um broker

### **Exceções:**

- `XMSEException`

- Exceção `IllegalState`
- `InvalidClientIDException`

*ExceptionListener - Obter e Configurar Listener de Exceção*

**Interface:**

```
ExceptionListener ExceptionListener
{
    get;
    set;
}
```

Obtenha o listener de exceção que está registrado com a conexão e registre um listener de exceção com a conexão.

Se nenhum listener de exceção for registrado com a conexão, o método retorna nulo. Se um listener de exceção já estiver registrado com a conexão, será possível cancelar o registro especificando um nulo em vez do listener de exceção.

Para obter mais informações sobre como usar listeners de exceções, consulte [Usando listeners de mensagens e de exceções no .NET](#)

**Exceções:**

- `XMSEException`

*Metadados-Obter metadados*

**Interface:**

```
IConnectionMetaData MetaData
{
    get;
}
```

Obter os metadados da conexão.

**Exceções:**

- `XMSEException`

**Métodos**

*Fechar-Fechar Conexão*

**Interface:**

```
void Close();
```

Feche a conexão.

Se um aplicativo tentar fechar uma conexão que já esteja fechada, a chamada será ignorada..

**Parâmetros:**

Nenhum

**Retorna:**

Nulo

**Exceções:**

- `XMSEException`

*CreateSession -Criar Sessão.*

**Interface:**

```
ISession CreateSession(Boolean transacted,  
                        AcknowledgeMode acknowledgeMode);
```

Criar uma sessão

**Parâmetros:**

**transacionado (entrada)**

O valor True significa que a sessão é transacionada O valor False significa que a sessão não foi transacionada.

Para uma conexão em tempo real com um broker, o valor deve ser False.

**acknowledgeMode (entrada).**

Indica como mensagens recebidas por um aplicativo são reconhecidas. O valor deve ser um dos seguintes do enumerador AcknowledgeMode :

```
AcknowledgeMode.AutoAcknowledge  
AcknowledgeMode.ClientAcknowledge  
AcknowledgeMode.DupsOkAcknowledge
```

Para uma conexão em tempo real com um broker, o valor deve ser AcknowledgeMode.AutoAcknowledge ou AcknowledgeMode.DupsOkAcknowledge

Esse parâmetro será ignorado se a sessão for transacionada.. Para obter mais informações sobre os modos de confirmação, consulte [Confirmação de mensagem](#)

**Retorna:**

O objeto Session

**Exceções:**

- XMSEException

*Iniciar-Iniciar Conexão*

**Interface:**

```
void Start();
```

Iniciar ou reiniciar a entrega de mensagens recebidas para a conexão. A chamada será ignorada se a conexão já tiver sido iniciada

**Parâmetros:**

Nenhum

**Retorna:**

Nulo

**Exceções:**

- XMSEException

*Parar-Parar Conexão*

**Interface:**

```
void Stop();
```

Pare a entrega de mensagens recebidas para a conexão. A chamada será ignorada se a conexão já tiver sido interrompida

**Parâmetros:**

Nenhum

**Retorna:**

Nulo

**Exceções:**

- XMSEException

**Propriedades e métodos herdados**

Os métodos a seguir são herdados da interface [IPropertyContext](#):

[GetBooleanProperty](#), [GetByteProperty](#), [GetBytesProperty](#), [GetCharProperty](#), [GetDoubleProperty](#), [GetFloatProperty](#), [GetIntProperty](#), [GetLongProperty](#), [GetObjectProperty](#), [GetShortProperty](#), [GetStringProperty](#), [SetBooleanProperty](#), [SetByteProperty](#), [SetBytesProperty](#), [SetCharProperty](#), [SetDoubleProperty](#), [SetFloatProperty](#), [SetIntProperty](#), [SetLongProperty](#), [SetObjectProperty](#), [SetShortProperty](#), [SetStringProperty](#)

**IConnectionFactory**

Um aplicativo usa um connection factory para criar uma conexão.

**Hierarquia de herança:**

```
IBM.XMS.IPropertyContext
|
+----IBM.XMS.IConnectionFactory
```

Para obter uma lista das propriedades definidas XMS de um objeto ConnectionFactory , consulte [“Propriedades de ConnectionFactory”](#) na página 2101.

**Métodos**

*CreateConnection* -Criar Connection Factory (utilizando a identidade do usuário padrão).

**Interface:**

```
IConnection CreateConnection();
```

Crie um connection factory com as propriedades padrão..

Se você estiver se conectando ao IBM MQ e XMSC\_USERID não estiver configurado, o gerenciador de filas usará o userID do usuário conectado por padrão. Se você precisar de autenticação no nível de conexão adicional de usuários individuais, poderá gravar uma saída de autenticação de cliente que está configurada em IBM MQ

**Parâmetros:**

Nenhum

**Exceções:**

- XMSEException

*CreateConnection* -Criar Conexão (usando uma identidade do usuário especificada).

**Interface:**

```
IConnection CreateConnection(String userId, String password);
```

Crie uma conexão usando uma identidade do usuário especificada

Se você estiver se conectando ao IBM MQ e XMSC\_USERID não estiver configurado, o gerenciador de filas usará o userID do usuário conectado por padrão. Se você precisar de autenticação no nível de conexão adicional de usuários individuais, poderá gravar uma saída de autenticação de cliente que está configurada em IBM MQ

A conexão é criada em modo interrompido Nenhuma mensagem é entregue até que o aplicativo chame **Connection.start()**

#### Parâmetros:

##### **userID (entrada)**

Um objeto String que encapsula o identificador de usuário a ser usado para autenticar o aplicativo.. Se você fornecer um valor nulo, será feita uma tentativa de criar a conexão sem autenticação.

##### **senha (entrada)**

Um objeto String encapsulando a senha a ser usada para autenticar o aplicativo. Se você fornecer um valor nulo, será feita uma tentativa de criar a conexão sem autenticação.

#### Retorna:

O objeto Conexão.

#### Exceções:

- XMSException
- XMS\_X\_SECURITY\_EXCEPTION

### **Propriedades e métodos herdados**

Os métodos a seguir são herdados da interface [IPropertyContext](#):

[GetBooleanProperty](#), [GetByteProperty](#), [GetBytesProperty](#), [GetCharProperty](#), [GetDoubleProperty](#), [GetFloatProperty](#), [GetIntProperty](#), [GetLongProperty](#), [GetObjectProperty](#), [GetShortProperty](#), [GetStringProperty](#), [SetBooleanProperty](#), [SetByteProperty](#), [SetBytesProperty](#), [SetCharProperty](#), [SetDoubleProperty](#), [SetFloatProperty](#), [SetIntProperty](#), [SetLongProperty](#), [SetObjectProperty](#), [SetShortProperty](#), [SetStringProperty](#)

### **IConnectionMetaDados**

Um objeto de dados ConnectionMeta fornece informações sobre uma conexão.

#### Hierarquia de herança:

```
IBM.XMS.IPropertyContext
|
+---- IBM.XMS.IConnectionMetaDados
```

Para obter uma lista das propriedades definidas pelo XMS de um objeto de dados ConnectionMeta, consulte [“Propriedades de Dados ConnectionMeta”](#) na página 2106

### **.NET propriedades**

*JMSXPropertyNames -Obter Propriedades de Mensagem Definidas JMS*

#### Interface:

```
System.Collections.IEnumerator JMSXPropertyNames
{
    get;
}
```

Retornar uma enumeração dos nomes das propriedades de mensagens definidas JMS suportadas pela conexão.

JMS propriedades de mensagem definidas não são suportadas por uma conexão em tempo real com um broker.

**Exceções:**

- XMSEException

**Propriedades e métodos herdados**

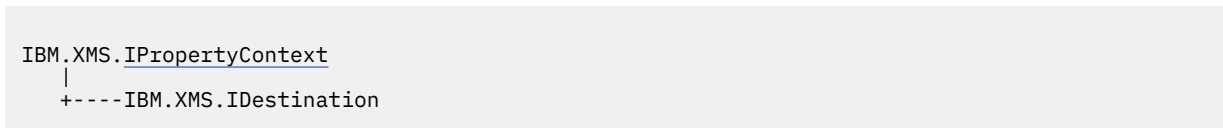
Os métodos a seguir são herdados da interface IPropertyContext:

GetBooleanProperty, GetByteProperty, GetBytesProperty, GetCharProperty, GetDoubleProperty, GetFloatProperty, GetIntProperty, GetLongProperty, GetObjectProperty, GetShortProperty, GetStringProperty, SetBooleanProperty, SetByteProperty, SetBytesProperty, SetCharProperty, SetDoubleProperty, SetFloatProperty, SetIntProperty, SetLongProperty, SetObjectProperty, SetShortProperty, SetStringProperty

**IDestination**

Um destino é para onde um aplicativo envia mensagens ou é uma origem da qual um aplicativo recebe mensagens, ou ambos.

**Hierarquia de herança:**



Para obter uma lista das propriedades definidas do XMS de um objeto de Destino, consulte [“Propriedades de Destino”](#) na página 2106

**.NET propriedades**

*Nome-Obter Nome do Destino*

**Interface:**

```
String Name
{
    get;
}
```

Obtenha o nome do destino O nome é uma sequência encapsulando o nome de uma fila ou o nome de um tópico..

**Exceções:**

- XMSEException

*TypeId -Obter Tipo de Destino*

**Interface:**

```
DestinationType TypeId
{
    get;
}
```

Obtenha o tipo do destino O tipo do destino é um dos seguintes valores:

DestinationType.Queue  
DestinationType.Topic

## Exceções:

- XMSException

## Propriedades e métodos herdados

Os métodos a seguir são herdados da interface IPropertyContext:

GetBooleanProperty, GetByteProperty, GetBytesProperty, GetCharProperty, GetDoubleProperty, GetFloatProperty, GetIntProperty, GetLongProperty, GetObjectProperty, GetShortProperty, GetStringProperty, SetBooleanProperty, SetByteProperty, SetBytesProperty, SetCharProperty, SetDoubleProperty, SetFloatProperty, SetIntProperty, SetLongProperty, SetObjectProperty, SetShortProperty, SetStringProperty

## ExceptionHandler

Um aplicativo usa um listener de exceção para ser notificado assincronamente de um problema com uma conexão.

### Hierarquia de herança:

Nenhum

Se um aplicativo usar uma conexão apenas para consumir mensagens de forma assíncrona e sem outro propósito, a única maneira de o aplicativo aprender sobre um problema com a conexão será usando um listener de exceção. Em outras situações, um listener de exceção pode fornecer uma maneira mais imediata de aprender sobre um problema com uma conexão do que esperar até a próxima chamada síncrona para XMS..

## Delegar

*ExceptionHandler -Listener de Exceção*

### Interface:

```
public delegate void ExceptionListener(Exception ex)
```

Notificar o aplicativo de um problema com uma conexão.

Os métodos que implementam esse delegado podem ser registrados com a conexão

Para obter mais informações sobre como usar listeners de exceções, consulte [Usando listeners de mensagens e de exceções no .NET](#)

### Parâmetros:

#### exceção (entrada)

Um ponteiro para uma exceção criada por XMS

### Retorna:

Nulo

## Exceção IllegalStateException

XMS lança essa exceção se um aplicativo chamar um método em um momento incorreto ou inadequado ou se XMS não estiver em um estado apropriado para a operação solicitada.

### Hierarquia de herança:

```
IBM.XMS.XMSException
|
+----IBM.XMS.Exception
|
+----IBM.XMS.IllegalStateException
```



## ***Propriedades e métodos herdados***

Os métodos a seguir são herdados da interface [XMSEException](#):

[GetErrorCódigo](#), [GetLinkedExceção](#)

## **InitialContext**

Um aplicativo usa um objeto InitialContext para criar objetos de definições de objeto que são recuperados de um repositório de objetos administrados.

### **Hierarquia de herança:**

Nenhum

## ***.NET propriedades***

*Ambiente-Obter o ambiente*

### **Interface:**

```
Hashtable Environment
{
    get;
}
```

Obter o ambiente

### **Exceções:**

- As exceções são específicas para o serviço de diretório que está sendo usado

## ***Construtores***

*InitialContext -Criar Contexto Inicial*

### **Interface:**

```
InitialContext(Hashtable env);
```

Crie um objeto InitialContext ..

### **Parâmetros:**

As informações necessárias para estabelecer uma conexão com o repositório de objetos administrados são fornecidas ao construtor em um ambiente Hashtable.

### **Exceções:**

- XMSEException

## ***Métodos***

*Ambiente AddTo-Incluir uma nova propriedade no ambiente*

### **Interface:**

```
Object AddToEnvironment(String propName, Object propVal);
```

Inclua uma nova propriedade no ambiente

### **Parâmetros:**

#### **propName (entrada)**

Um objeto String encapsulando o nome da propriedade a ser incluída.

**propVal (entrada)**

O valor da propriedade a ser incluída.

**Retorna:**

O valor antigo da propriedade.

**Exceções:**

- As exceções são específicas para o serviço de diretório que está sendo usado

*Fechar-Fechar este contexto*

**Interface:**

```
void Close()
```

Feche este contexto.

**Parâmetros:**

Nenhum

**Retorna:**

Nenhum

**Exceções:**

- As exceções são específicas para o serviço de diretório que está sendo usado

*Consulta-objeto de consulta no contexto inicial*

**Interface:**

```
Object Lookup(String name);
```

Crie um objeto a partir de uma definição de objeto que é recuperado do repositório de objetos administrados.

**Parâmetros:****nome (entrada)**

Um objeto String encapsulando o nome do objeto administrado a ser recuperado. O nome pode ser um nome simples ou um nome complexo. Para obter detalhes adicionais, consulte [Recuperação de objetos administradas](#)

**Retorna:**

Um IConnectionFactory ou um IDestination, dependendo do tipo de objeto sendo recuperado. Se a função puder acessar o diretório, mas não puder localizar o objeto necessário, um nulo será retornado..

**Exceções:**

- As exceções são específicas para o serviço de diretório que está sendo usado

*Ambiente RemoveFrom-Remover uma Propriedade do Ambiente.*

**Interface:**

```
Object RemoveFromEnvironment(String propName);
```

Remova uma propriedade do ambiente.

**Parâmetros:****propName (entrada)**

Um objeto String encapsulando o nome da propriedade a ser removida.

**Retorna:**

O objeto que foi removido..

**Exceções:**

- As exceções são específicas para o serviço de diretório que está sendo usado

## InvalidClientIDException

XMS lança essa exceção se um aplicativo tentar configurar um identificador de cliente para uma conexão, mas o identificador de cliente não for válido ou já estiver em uso.

**Hierarquia de herança:**

```
IBM.XMS.XMSEException
|
+----IBM.XMS.XMSEException
      |
      +----IBM.XMS.InvalidClientIDException
```

**Propriedades e métodos herdados**

Os métodos a seguir são herdados da interface [XMSEException](#):

[GetErrorCódigo](#), [GetLinkedExceção](#)

## Exceção de InvalidDestination

XMS lançará essa exceção se um aplicativo especificar um destino que não seja válido

**Hierarquia de herança:**

```
IBM.XMS.XMSEException
|
+----IBM.XMS.XMSEException
      |
      +----IBM.XMS.InvalidDestinationException
```

**Propriedades e métodos herdados**

Os métodos a seguir são herdados da interface [XMSEException](#):

[GetErrorCódigo](#), [GetLinkedExceção](#)

## InvalidSelectorExceção

XMS emitirá essa exceção se um aplicativo fornecer uma expressão de seletor de mensagem cuja sintaxe não é válida

**Hierarquia de herança:**

```
IBM.XMS.XMSEException
|
+----IBM.XMS.XMSEException
      |
      +----IBM.XMS.InvalidSelectorException
```

**Propriedades e métodos herdados**

Os métodos a seguir são herdados da interface [XMSEException](#):

[GetErrorCódigo](#), [GetLinkedExceção](#)

## IMapMessage

Uma mensagem de mapa é uma mensagem cujo corpo consiste em um conjunto de pares nome-valor, em que cada valor possui um tipo de dados associado.

### Hierarquia de herança:

```
IBM.XMS.IPropertyContext
|
+----IBM.XMS.IMessage
|
+----IBM.XMS.IMapMessage
```

Quando um aplicativo obtém o valor do par nome-valor, o valor pode ser convertido por XMS em outro tipo de dado.. Para obter mais informações sobre esta forma de conversão implícita, consulte as informações sobre mensagens de mapa em [O corpo de uma mensagem do XMS](#)

### .NET propriedades

*MapNames* -Obter nomes de mapa

#### Interface:

```
System.Collections.IEnumerator MapNames
{
    get;
}
```

Obter uma enumeração dos nomes no corpo da mensagem de mapa.

#### Exceções:

- XMSEException

### Métodos

*GetBoolean* -Obter Valor Booleano

#### Interface:

```
Boolean GetBoolean(String name);
```

Obtenha o valor booleano identificado pelo nome do corpo da mensagem do mapa.

#### Parâmetros:

##### nome (entrada)

Um objeto String encapsulando o nome que identifica o valor booleano.

#### Retorna:

O valor booleano recuperado do corpo da mensagem de mapa.

#### Exceções:

- XMSEException

*GetByte* -Obter Byte.

#### Interface:

```
Byte    GetByte(String name);
Int16   GetSignedByte(String name);
```

Obtenha o byte identificado por nome a partir do corpo da mensagem do mapa

**Parâmetros:****nome (entrada)**

Um objeto String encapsulando o nome que identifica o byte.

**Retorna:**

O byte recuperado do corpo da mensagem de mapa. Nenhuma conversão de dados é executada no byte..

**Exceções:**

- XMSEException

*GetBytes -Obter Bytes*

**Interface:**

```
Byte[] GetBytes(String name);
```

Obtenha a matriz de bytes identificados por nome a partir do corpo da mensagem do mapa

**Parâmetros:****nome (entrada)**

Um objeto String encapsulando o nome que identifica a matriz de bytes..

**Retorna:**

O número de bytes na matriz

**Exceções:**

- XMSEException

*GetChar -Obter Caractere*

**Interface:**

```
Char GetChar(String name);
```

Obter o caractere identificado pelo nome a partir do corpo da mensagem do mapa

**Parâmetros:****nome (entrada)**

Um objeto String encapsulando o nome que identifica o caractere.

**Retorna:**

O caractere recuperado do corpo da mensagem do mapa.

**Exceções:**

- XMSEException

*GetDouble -Obter Número de Ponto Flutuante de Precisão Dupla*

**Interface:**

```
Double GetDouble(String name);
```

Obter o número de vírgula flutuante de precisão dupla identificado por nome a partir do corpo da mensagem do mapa

**Parâmetros:****nome (entrada)**

Um objeto String encapsulando o nome que identifica o número de vírgula flutuante de precisão dupla.

**Retorna:**

O número de vírgula flutuante de dupla precisão recuperado do corpo de mensagem do mapa

**Exceções:**

- XMSEException

*GetFloat -Obter número de ponto flutuante*

**Interface:**

```
Single GetFloat(String name);
```

Obtenha o número de vírgula flutuante identificado por nome a partir do corpo da mensagem do mapa

**Parâmetros:****nome (entrada)**

Um objeto String encapsulando o nome que identifica o número do ponto flutuante.

**Retorna:**

O número de ponto flutuante recuperado do corpo da mensagem do mapa

**Exceções:**

- XMSEException

*GetInt -Obter Número Inteiro*

**Interface:**

```
Int32 GetInt(String name);
```

Obter o número inteiro identificado por nome do corpo da mensagem do mapa.

**Parâmetros:****nome (entrada)**

Um objeto String encapsulando o nome que identifica o número inteiro.

**Retorna:**

O número inteiro recuperado do corpo da mensagem de mapa.

**Exceções:**

- XMSEException

*GetLong -Obter Número Inteiro Longo*

**Interface:**

```
Int64 GetLong(String name);
```

Obter o número inteiro longo identificado pelo nome do corpo da mensagem do mapa.

**Parâmetros:****nome (entrada)**

Um objeto String encapsulando o nome que identifica o número inteiro longo.

**Retorna:**

O número inteiro longo recuperado do corpo da mensagem do mapa.

**Exceções:**

- XMSEException

*GetObject -Obter Objeto*

**Interface:**

```
Object GetObject(String name);
```

Obter uma referência para o valor de um par nome-valor a partir do corpo da mensagem do mapa. O par de nome-valor é identificado por nome.

**Parâmetros:**

**nome (entrada)**

Um objeto String encapsulando o nome do par nome-valor.

**Retorna:**

O valor, que é um dos seguintes tipos de objeto:

Boolean  
Byte  
Byte[]  
Char  
Double  
Single  
Int32  
Int64  
Int16  
String

**Exceções:**

XMSEException

*GetShort -Obter Número Inteiro Curto*

**Interface:**

```
Int16 GetShort(String name);
```

Obter o número inteiro curto identificado por nome a partir do corpo da mensagem do mapa.

**Parâmetros:**

**nome (entrada)**

Um objeto String encapsulando o nome que identifica o número inteiro curto.

**Retorna:**

O número inteiro curto recuperado do corpo da mensagem do mapa.

**Exceções:**

- XMSEException

*GetString -Obter sequência*

**Interface:**

```
String GetString(String name);
```

Obtenha a sequência identificada pelo nome a partir do corpo da mensagem do mapa

**Parâmetros:**

**nome (entrada)**

Um objeto String encapsulando o nome que identifica a sequência no corpo da mensagem do mapa.

**Retorna:**

Um objeto String encapsulando a sequência recuperada do corpo de uma mensagem de mapa. Se a conversão de dados for necessária, esse valor será a cadeia após a conversão..

**Exceções:**

- XMSEException

*ItemExists* -Verifique se o par nome-valor existe

**Interface:**

```
Boolean ItemExists(String name);
```

Verifique se o corpo da mensagem de mapa contém um par nome-valor com o nome especificado.

**Parâmetros:****nome (entrada)**

Um objeto String encapsulando o nome do par nome-valor.

**Retorna:**

- True, se o corpo da mensagem de mapa contiver um par nome-valor com o nome especificado.
- False, se o corpo da mensagem de mapa não contiver um par nome-valor com o nome especificado.

**Exceções:**

- XMSEException

*SetBoolean* -Configurar Valor Booleano

**Interface:**

```
void SetBoolean(String name, Boolean value);
```

Configure um valor booleano no corpo da mensagem de mapa.

**Parâmetros:****nome (entrada)**

Um objeto String encapsulando o nome para identificar o valor booleano no corpo da mensagem do mapa..

**valor (entrada)**

O valor booleano a ser configurado.

**Retorna:**

Nulo

**Exceções:**

- XMSEException

*SetByte* -Configurar Byte

**Interface:**

```
void SetByte(String name, Byte value);  
void SetSignedByte(String name, Int16 value);
```

Configure um byte no corpo da mensagem do mapa



**Parâmetros:****nome (entrada)**

Um objeto String encapsulando o nome para identificar o byte no corpo de uma mensagem de mapa

**valor (entrada)**

O byte a ser configurado..

**Retorna:**

Nulo

**Exceções:**

- XMSEException

*SetBytes -Configurar Bytes*

**Interface:**

```
void SetBytes(String name, Byte[] value);
```

Configure uma matriz de bytes no corpo da mensagem de mapa

**Parâmetros:****nome (entrada)**

Um objeto String encapsulando o nome para identificar a matriz de bytes no corpo da mensagem do mapa

**valor (entrada)**

A matriz de bytes a ser configurada

**Retorna:**

Nulo

**Exceções:**

- XMSEException

*SetChar -Configurar Caractere*

**Interface:**

```
void SetChar(String name, Char value);
```

Configure um caractere de 2 bytes no corpo da mensagem de mapeamento.

**Parâmetros:****nome (entrada)**

Um objeto String encapsulando o nome para identificar o caractere no corpo da mensagem do mapa

**valor (entrada)**

O caractere a ser configurado.

**Retorna:**

Nulo

**Exceções:**

- XMSEException

## *SetDouble -Configurar Número de Ponto Flutuante de Precisão Dupla*

### **Interface:**

```
void SetDouble(String name, Double value);
```

Configure um número de ponto flutuante de precisão dupla no corpo da mensagem do mapa.

### **Parâmetros:**

#### **nome (entrada)**

Um objeto String encapsulando o nome para identificar o número do ponto flutuante de precisão dupla no corpo da mensagem do mapa

#### **valor (entrada)**

O número do ponto flutuante de precisão dupla a ser configurado

### **Retorna:**

Nulo

### **Exceções:**

- XMSEException

## *SetFloat -Configurar número de ponto flutuante*

### **Interface:**

```
void SetFloat(String name, Single value);
```

Configure um número de vírgula flutuante no corpo da mensagem de mapa

### **Parâmetros:**

#### **nome (entrada)**

Um objeto String encapsulando o nome para identificar o número de ponto flutuante no corpo da mensagem do mapa

#### **valor (entrada)**

O número de vírgula flutuante a ser configurado

### **Retorna:**

Nulo

### **Exceções:**

- XMSEException

## *SetInt -Configurar Número Inteiro*

### **Interface:**

```
void SetInt(String name, Int32 value);
```

Configure um número inteiro no corpo da mensagem do mapa.

### **Parâmetros:**

#### **nome (entrada)**

Um objeto String encapsulando o nome para identificar o número inteiro no corpo da mensagem do mapa.

#### **valor (entrada)**

O número inteiro a ser configurado.

### **Retorna:**

Nulo

**Exceções:**

- XMSEException

*SetLong -Configurar número inteiro longo*

**Interface:**

```
void SetLong(String name, Int64 value);
```

Configure um número inteiro longo no corpo da mensagem de mapa

**Parâmetros:****nome (entrada)**

Um objeto String encapsulando o nome para identificar o número inteiro longo no corpo da mensagem do mapa..

**valor (entrada)**

O número inteiro longo a ser configurado.

**Retorna:**

Nulo

**Exceções:**

- XMSEException

*SetObject -Configurar Objeto*

**Interface:**

```
void SetObject(String name, Object value);
```

Configure um valor, que deve ser um tipo primitivo XMS , no corpo da mensagem do mapa.

**Parâmetros:****nome (entrada)**

Um objeto String encapsulando o nome para identificar o valor no corpo da mensagem do mapa.

**valor (entrada)**

Uma matriz de bytes contendo o valor a ser configurado.

**Retorna:**

Nulo

**Exceções:**

- XMSEException

*SetShort -Configurar Número Inteiro Curto*

**Interface:**

```
void SetShort(String name, Int16 value);
```

Configure um número inteiro curto no corpo da mensagem do mapa.

**Parâmetros:****nome (entrada)**

Um objeto String encapsulando o nome para identificar o número inteiro curto no corpo da mensagem do mapa.

**valor (entrada)**

O número inteiro curto a ser configurado.

**Retorna:**

Nulo

**Exceções:**

- XMSEException

*SetString -Configurar Sequência***Interface:**

```
void SetString(String name, String value);
```

Configure uma sequência no corpo da mensagem do mapa.

**Parâmetros:****nome (entrada)**

Um objeto String encapsulando o nome para identificar a sequência no corpo da mensagem do mapa.

**valor (entrada)**

Um objeto String encapsulando a sequência a ser configurada.

**Retorna:**

Nulo

**Exceções:**

- XMSEException

**Propriedades e métodos herdados**As propriedades a seguir são herdadas da interface IMessage:

JMSCorrelationID, JMSDeliveryMode, JMSDestination, JMSExpiration, JMSMessageID, JMSPriority, JMSRedelivered, JMSReplyTo, JMSTimestamp, JMSType, Propriedades

Os métodos a seguir são herdados da interface IMessage:

clearBody, clearProperties, PropertyExists

Os métodos a seguir são herdados da interface IPropertyContext:

GetBooleanProperty, GetByteProperty, GetBytesProperty, GetCharProperty, GetDoubleProperty, GetFloatProperty, GetIntProperty, GetLongProperty, GetObjectProperty, GetShortProperty, GetStringProperty, SetBooleanProperty, SetByteProperty, SetBytesProperty, SetCharProperty, SetDoubleProperty, SetFloatProperty, SetIntProperty, SetLongProperty, SetObjectProperty, SetShortProperty, SetStringProperty

**IMessage**

Um objeto de mensagem representa uma mensagem que um aplicativo envia ou recebe. IMessage é uma superclasse para as classes de mensagem, como IMapMessage.

**Hierarquia de herança:**

```
IBM.XMS.IPropertyContext
|
+---- IBM.XMS.IMessage
```

Para obter uma lista dos campos de cabeçalho da mensagem JMS em um objeto de Mensagem, consulte Campos de cabeçalho de uma mensagem XMS. Para obter uma lista das propriedades definidas JMS de um objeto de Mensagem, consulte Propriedades definidas pelo JMS de uma mensagem. Para obter uma lista das propriedades definidas pelo IBM de um objeto de Mensagem, consulte IBM de uma mensagem. Para obter uma lista de propriedades JMS\_IBM\_MQMD\* para o objeto Message, consulte “Propriedades JMS\_IBM\_MQMD\*” na página 2110

As mensagens são excluídas pelo coletor de lixo Quando uma mensagem é excluída, isso libera os recursos que ela estava usando.

## **.NET propriedades**

*GetJMSCorrelationID-Obter e Configurar JMSCorrelationID*

### **Interface:**

```
String JMSCorrelationID
{
    get;
    set;
}
```

Obtenha e configure o identificador de correlação da mensagem como um objeto de Sequência.

### **Exceções:**

- XMSEException

*JMSDeliveryMode -Obter e configurar JMSDeliveryMode*

### **Interface:**

```
DeliveryMode JMSDeliveryMode
{
    get;
    set;
}
```

Obter e configurar o modo de entrega da mensagem.

O modo de entrega da mensagem é um dos seguintes valores:

```
DeliveryMode.Persistent
DeliveryMode.NonPersistent
```

Para uma mensagem recém-criada que não foi enviada, o modo de entrega é `DeliveryMode.Persistente`, exceto para uma conexão em tempo real com um broker para o qual o modo de entrega é `DeliveryMode.NonPersistent` Para uma mensagem recebida, o método retorna o modo de entrega que foi configurado pela chamada `IMessageProducer.send ()` quando a mensagem foi enviada, a menos que o aplicativo de recebimento altere o modo de entrega configurando `JMSDeliveryMode`.

### **Exceções:**

- XMSEException

*JMSDestination-Obter e Configurar JMSDestination.*

### **Interface:**

```
IDestination JMSDestination
{
    get;
    set;
}
```

Obtenha e configure o destino da mensagem

O destino é configurado pela chamada de `IMessageProducer.send ()` quando a mensagem é enviada. O valor de `JMSDestination` é ignorado. No entanto, é possível usar `JMSDestination` para alterar o destino de uma mensagem recebida.

Para uma mensagem recém-criada que não foi enviada, o método retorna um objeto Destino nulo, a menos que o aplicativo de envio configure um destino configurando JMSDestination. Para uma mensagem recebida, o método retorna um objeto Destination para o destino que foi configurado pela chamada IMessageProducer.send () quando a mensagem foi enviada, a menos que o aplicativo de recebimento altere o destino configurando JMSDestination.

**Exceções:**

- XMSEException

*JMSExpiration-Obter e configurar JMSExpiration*

**Interface:**

```
Int64 JMSExpiration
{
    get;
    set;
}
```

Obter e configurar o prazo de expiração da mensagem

O prazo de expiração é configurado pela chamada de IMessageProducer.send () quando a mensagem é enviada. Seu valor é calculado adicionando o tempo de vida, conforme especificado pelo aplicativo de envio, ao tempo em que a mensagem é enviada. O prazo de expiração é expresso em milissegundos desde 00:00:00 GMT de 1 de janeiro de 1970.

Para uma mensagem recém-criada que não foi enviada, o prazo de expiração é 0, a menos que o aplicativo de envio configure um prazo de expiração diferente configurando JMSExpiration. Para uma mensagem recebida, o método retorna o prazo de expiração que foi configurado pela chamada IMessageProducer.send () quando a mensagem foi enviada, a menos que o aplicativo de recebimento altere o prazo de expiração configurando JMSExpiration.

Se o tempo de vida for 0, a chamada IMessageProducer.send () configurará o tempo de vencimento como 0 para indicar que a mensagem não expira...

O XMS descarta mensagens expiradas e não as entrega para aplicativos.

**Exceções:**

- XMSEException

*JMSMessageID -Obter e configurar JMSMessageID*

**Interface:**

```
String JMSMessageID
{
    get;
    set;
}
```

Obtenha e configure o identificador de mensagem da mensagem como um objeto de sequência encapsulando o identificador de mensagens.

O identificador de mensagem é configurado pela chamada .send () de IMessageProducer quando a mensagem é enviada. Para uma mensagem que foi recebida, o método retorna o identificador de mensagem que foi configurado pela chamada IMessageProducer.send () quando a mensagem foi enviada, a menos que o aplicativo de recebimento altere o identificador de mensagem configurando JMSMessageID...

Se a mensagem não tiver identificador de mensagem, o método retornará um nulo.

**Exceções:**

- XMSEException

*JMSPriority-Obter e configurar JMSPriority.*

**Interface:**

```
Int32 JMSPriority
{
    get;
    set;
}
```

Obtenha e configure a prioridade da mensagem

A prioridade é configurada pela chamada `IMessageProducer.send ()` quando a mensagem é enviada O valor é um número inteiro no intervalo 0, a prioridade mais baixa, para 9, a prioridade mais alta.

Para uma mensagem recém-criada que não foi enviada, a prioridade será 4 a menos que o aplicativo de envio configure uma prioridade diferente configurando `JMSPriority`. Para uma mensagem recebida, o método retorna a prioridade que foi configurada pela chamada `IMessageProducer.send ()` quando a mensagem foi enviada, a menos que o aplicativo de recebimento altere a prioridade configurando `JMSPriority`.

**Exceções:**

- `XMSEException`

*JMSRedelivered-Obter e Configurar JMSRedelivered*

**Interface:**

```
Boolean JMSRedelivered
{
    get;
    set;
}
```

Obter uma indicação se a mensagem está sendo entregue novamente e indicar se a mensagem está sendo entregue novamente. A indicação é configurada pela chamada `IMessageConsumer.receive ()` quando a mensagem é recebida.

Essa propriedade possui os seguintes valores:

- `True`, se a mensagem estiver sendo entregue novamente
- `False`, se a mensagem não estiver sendo entregue novamente

Para uma conexão em tempo real com um broker, o valor é sempre `False`

Uma indicação de nova entrega configurada por `JMSRedelivered` antes de a mensagem ser enviada é ignorada pela chamada `IMessageProducer.send ()` quando a mensagem é enviada e é ignorada e substituída pela chamada `IMessageConsumer.receive ()` quando a mensagem é recebida. No entanto, é possível usar `JMSRedelivered` para alterar a indicação de uma mensagem recebida.

**Exceções:**

- `XMSEException`

*JMSReplyTo -Get e Configurar JMSReplyTo*

**Interface:**

```
IDestination JMSReplyTo
{
    get;
    set;
}
```

Obter e configurar o destino para o qual uma resposta à mensagem deve ser enviada.

O valor dessa propriedade é um objeto de Destino para o destino no qual uma resposta à mensagem deve ser enviada. Um objeto Destino nulo significa que nenhuma resposta é esperada.

**Exceções:**

- XMSEException

*JMSTimestamp-Obter e configurar JMSTimestamp.*

**Interface:**

```
Int64 JMSTimestamp
{
    get;
    set;
}
```

Obter e configurar o horário em que a mensagem foi enviada

O registro de data e hora é configurado pela chamada `IMessageProducer.send ()` quando a mensagem é enviada e é expressa em milissegundos desde 00:00:00 GMT de 1 de janeiro de 1970.

Para uma mensagem recém-criada que não foi enviada, o registro de data e hora será 0, a menos que o aplicativo de envio configure um registro de data e hora diferente configurando `JMSTimestamp`. Para uma mensagem que foi recebida, o método retorna o registro de data e hora que foi configurado pela chamada `IMessageProducer.send ()` quando a mensagem foi enviada, a menos que o aplicativo de recebimento altere o registro de data e hora configurando `JMSTimestamp`.

**Exceções:**

- XMSEException

**Notas:**

1. Se o registro de data e hora for indefinido, o método retornará 0, mas não lançará nenhuma exceção

*JMSType-Obter e Configurar JMSType*

**Interface:**

```
String JMSType
{
    get;
    set;
}
```

Get e configure o tipo da mensagem.

O valor de `JMSType` é uma sequência encapsulando o tipo da mensagem... Se a conversão de dados for necessária, esse valor será o tipo após a conversão..

**Exceções:**

- XMSEException

*PropertyNames -Obter Propriedades*

**Interface:**

```
System.Collections.IEnumerator PropertyNames
{
    get;
}
```

Obter uma enumeração das propriedades de nomes da mensagem



**Exceções:**

- XMSEException

**Métodos***Reconhecimento-Reconhecimento***Interface:**

```
void Acknowledge();
```

Reconheça essa mensagem e todas as mensagens não reconhecidas anteriormente recebidas pela sessão.

Um aplicativo pode chamar esse método se o modo de confirmação da sessão for AcknowledgeModeClientAcknowledge. As chamadas para o método serão ignoradas se a sessão tiver qualquer outro modo de reconhecimento ou for transacionada

As mensagens que foram recebidas, mas não reconhecidas, podem ser entregues novamente.

Para obter mais informações sobre o reconhecimento de mensagens, consulte [../develop/xms\\_cmescack.dita#xms\\_cmescack](http://develop/xms_cmescack.dita#xms_cmescack)

**Parâmetros:**

Nenhum

**Retorna:**

Nulo

**Exceções:**

- XMSEException
- Exceção IllegalStateException

*ClearBody -Limpar corpo***Interface:**

```
void ClearBody();
```

Limpe o corpo da mensagem. Os campos de cabeçalho e as propriedades de mensagem não são limpos

Se um aplicativo limpar um corpo da mensagem, o corpo permanecerá no mesmo estado que um corpo vazio em uma mensagem recém-criada O estado de um corpo vazio em uma mensagem recém-criada depende do tipo de corpo da mensagem.. Para obter mais informações, consulte [O corpo de uma mensagem do XMS](#)

Um aplicativo pode limpar um corpo de mensagem a qualquer momento, não importa em qual estado o corpo está. Se um corpo da mensagem for somente leitura, a única maneira de um aplicativo poder gravar no corpo será para o aplicativo limpar o corpo primeiro.

**Parâmetros:**

Nenhum

**Retorna:**

Nulo

**Exceções:**

- XMSEException

## *ClearProperties -Limpar Propriedades*

### **Interface:**

```
void ClearProperties();
```

limpa as propriedades da mensagem. Os campos de cabeçalho e o corpo da mensagem não são limpos. Se um aplicativo limpar as propriedades de uma mensagem, as propriedades se tornarão legíveis e graváveis.

Um aplicativo pode limpar as propriedades de uma mensagem a qualquer momento, independentemente do estado em que as propriedades estão. Se as propriedades de uma mensagem forem somente leitura, a única maneira de as propriedades se tornarem graváveis será para o aplicativo limpar as propriedades primeiro.

### **Parâmetros:**

Nenhum

### **Retorna:**

Nulo

### **Exceções:**

- XMSEException

## *PropertyExists -Verificar propriedade existe*

### **Interface:**

```
Boolean PropertyExists(String propertyName);
```

Verifique se a mensagem possui uma propriedade com o nome especificado.

### **Parâmetros:**

#### **propertyName (entrada)**

Um objeto String encapsulando o nome da propriedade.

### **Retorna:**

- True, se a mensagem tiver uma propriedade com o nome especificado
- False, se a mensagem não tiver uma propriedade com o nome especificado

### **Exceções:**

- XMSEException

## ***Propriedades e métodos herdados***

Os métodos a seguir são herdados da interface [IPropertyContext](#):

[GetBooleanProperty](#), [GetByteProperty](#), [GetBytesProperty](#), [GetCharProperty](#), [GetDoubleProperty](#), [GetFloatProperty](#), [GetIntProperty](#), [GetLongProperty](#), [GetObjectProperty](#), [GetShortProperty](#), [GetStringProperty](#), [SetBooleanProperty](#), [SetByteProperty](#), [SetBytesProperty](#), [SetCharProperty](#), [SetDoubleProperty](#), [SetFloatProperty](#), [SetIntProperty](#), [SetLongProperty](#), [SetObjectProperty](#), [SetShortProperty](#), [SetStringProperty](#)

## **IMessageConsumer**

Um aplicativo usa um consumidor de mensagem para receber mensagens enviadas a um destino.

### **Hierarquia de herança:**

```
IBM.XMS.IPropertyContext
```

```
|  
+----IBM.XMS.IMessageConsumer
```

Para obter uma lista das propriedades definidas do XMS de um objeto MessageConsumer , consulte [“Propriedades de MessageConsumer”](#) na página 2114

## **.NET propriedades**

*MessageListener -Obter e Configurar Listener de Mensagens*

### **Interface:**

```
MessageListener MessageListener  
{  
    get;  
    set;  
}
```

Obter o listener de mensagens que está registrado com o consumidor de mensagem e registrar um listener de mensagens com o consumidor de mensagem.

Se nenhum listener de mensagens estiver registrado com o consumidor de mensagens, MessageListener será nulo. Se um listener de mensagem já estiver registrado com o consumidor de mensagens, será possível cancelar o registro especificando um nulo.

Para obter mais informações sobre como usar listeners de mensagens, consulte [Usando listeners de mensagem e de exceção em .NET](#).

### **Exceções:**

- XMSEException

*MessageSelector -Obter Seletor de Mensagem*

### **Interface:**

```
String MessageSelector  
{  
    get;  
}
```

Obter o seletor de mensagem para o consumidor de mensagens O valor de retorno é um objeto String encapsulando a expressão do seletor de mensagem. Se a conversão de dados for necessária, esse valor será a expressão do seletor de mensagem após a conversão.. Se o consumidor de mensagens não tiver um seletor de mensagem, o valor de MessageSelector será um objeto String nulo.

### **Exceções:**

- XMSEException

## **Métodos**

*Fechar-Fechar Consumidor da Mensagem.*

### **Interface:**

```
void Close();
```

Feche o consumidor de mensagens.

Se um aplicativo tentar fechar um consumidor de mensagens que já esteja encerrado, a chamada será ignorada

**Parâmetros:**

Nenhum

**Retorna:**

Nulo

**Exceções:**

- XMSEException

*Receber-Receber*

**Interface:**

```
IMessage Receive();
```

Receber a próxima mensagem para o consumidor de mensagens A chamada espera indefinidamente por uma mensagem ou até o consumidor de mensagem ser fechado.

**Parâmetros:**

Nenhum

**Retorna:**

Um ponteiro para o objeto de Mensagem Se o consumidor de mensagens for fechado enquanto a chamada estiver aguardando uma mensagem, o método retornará um ponteiro para um objeto Mensagem nulo.

**Exceções:**

- XMSEException

*Recebimento-Recebimento (com um intervalo de espera)*

**Interface:**

```
IMessage Receive(Int64 delay);
```

Receber a próxima mensagem para o consumidor de mensagens A chamada aguarda apenas um período especificado para uma mensagem ou até que o consumidor de mensagens seja fechado.

**Parâmetros:****atraso (entrada)**

O tempo, em milissegundos, que a chamada espera por uma mensagem Se você especificar um intervalo de espera de 0, a chamada aguardará indefinidamente uma mensagem.

**Retorna:**

Um ponteiro para o objeto de Mensagem Se nenhuma mensagem chegar durante o intervalo de espera ou se o consumidor de mensagens for fechado enquanto a chamada estiver aguardando uma mensagem, o método retornará um ponteiro para um objeto de Mensagem nulo, mas não emitirá nenhuma exceção.

**Exceções:**

- XMSEException

*ReceiveNoEspera-Receber com Nenhuma Espera*

**Interface:**

```
IMessage ReceiveNoWait();
```

Receba a próxima mensagem para o consumidor de mensagens se uma estiver disponível imediatamente.

**Parâmetros:**

Nenhum

**Retorna:**

Um ponteiro para um objeto de Mensagem Se nenhuma mensagem estiver disponível imediatamente, o método retornará um ponteiro para um objeto Mensagem nulo.

**Exceções:**

- XMSEException

**Propriedades e métodos herdados**

Os métodos a seguir são herdados da interface [IPropertyContext](#):

[GetBooleanProperty](#), [GetByteProperty](#), [GetBytesProperty](#), [GetCharProperty](#), [GetDoubleProperty](#), [GetFloatProperty](#), [GetIntProperty](#), [GetLongProperty](#), [GetObjectProperty](#), [GetShortProperty](#), [GetStringProperty](#), [SetBooleanProperty](#), [SetByteProperty](#), [SetBytesProperty](#), [SetCharProperty](#), [SetDoubleProperty](#), [SetFloatProperty](#), [SetIntProperty](#), [SetLongProperty](#), [SetObjectProperty](#), [SetShortProperty](#), [SetStringProperty](#)

**MessageEOFException**

XMS lança essa exceção se XMS encontrar o final de um fluxo de mensagens de bytes quando um aplicativo estiver lendo o corpo de uma mensagem de bytes.

**Hierarquia de herança:**

```

IBM.XMS.XMSEException
|
+----IBM.XMS.XMSEException
      |
      +----IBM.XMS.MessageEOFException
  
```

**Propriedades e métodos herdados**

Os métodos a seguir são herdados da interface [XMSEException](#):

[GetErrorCódigo](#), [GetLinkedExceção](#)

**MessageFormatException**

XMS lançará essa exceção se o XMS encontrar uma mensagem com um formato que não seja válido

**Hierarquia de herança:**

```

IBM.XMS.XMSEException
|
+----IBM.XMS.XMSEException
      |
      +----IBM.XMS.MessageFormatException
  
```

**Propriedades e métodos herdados**

Os métodos a seguir são herdados da interface [XMSEException](#):

[GetErrorCódigo](#), [GetLinkedExceção](#)

**IMessageListener (delegado)**

Um aplicativo usa um listener de mensagens para receber mensagens assincronamente.

**Hierarquia de herança:**

Nenhum

**Delegar**

## MessageListener -Listener de Mensagens

### Interface:

```
public delegate void MessageListener(IMessage msg);
```

Entregar uma mensagem de forma assíncrona ao consumidor de mensagem.

Os métodos que implementam esse delegado podem ser registrados com a conexão

Para obter mais informações sobre como usar listeners de mensagem, consulte [Usando listeners de mensagem e de exceção no .NET](#).

### Parâmetros:

#### **mesg (entrada)**

O objeto de Mensagem

### Retorna:

Nulo

## MessageNotReadableException

XMS lança essa exceção se um aplicativo tentar ler o corpo de uma mensagem que é somente gravação.

### Hierarquia de herança:

```
IBM.XMS.XMSEException
|
+----IBM.XMS.XMSEException
|
+----IBM.XMS.MessageNotReadableException
```

### Propriedades e métodos herdados

Os métodos a seguir são herdados da interface [XMSEException](#):

[GetErrorCódigo](#), [GetLinkedExceção](#)

## MessageNotWritableException

XMS lança essa exceção se um aplicativo tentar gravar no corpo de uma mensagem que é somente leitura.

### Hierarquia de herança:

```
IBM.XMS.XMSEException
|
+----IBM.XMS.XMSEException
|
+----IBM.XMS.MessageNotWritableException
```

### Propriedades e métodos herdados

Os métodos a seguir são herdados da interface [XMSEException](#):

[GetErrorCódigo](#), [GetLinkedExceção](#)

## IMessageProducer

Um aplicativo usa um produtor de mensagem para enviar mensagens para um destino.

### Hierarquia de herança:

```
IBM.XMS.IPropertyContext
```

```
|  
+----IBM.XMS.IMessageProducer
```

Para obter uma lista das propriedades definidas XMS de um objeto MessageProducer , consulte [“Propriedades do MessageProducer” na página 2114.](#)

## **.NET propriedades**

*DeliveryMode -Obter e Configurar Modo de Entrega Padrão*

### **Interface:**

```
DeliveryMode DeliveryMode  
{  
    get;  
    set;  
}
```

Obter e configurar o modo de entrega padrão para mensagens enviadas pelo produtor de mensagens

O modo de entrega padrão possui um dos seguintes valores:

```
DeliveryMode.Persistent  
DeliveryMode.NonPersistent
```

Para uma conexão em tempo real com um broker, o valor deve ser `DeliveryMode.NonPersistent`.

O valor padrão é `DeliveryMode.Persistent`, exceto para uma conexão em tempo real com um broker para o qual o valor padrão é `DeliveryMode.NonPersistent`.

### **Exceções:**

- `XMSEException`

*Destino-Obter Destino*

### **Interface:**

```
IDestination Destination  
{  
    get;  
}
```

Obter o destino para o produtor de mensagem

### **Parâmetros:**

Nenhum

### **Retorna:**

O objeto de Destino Se o produtor da mensagem não tiver um destino, o método retornará um objeto de Destino nulo...

### **Exceções:**

- `XMSEException`

*ID de DisableMsg-Obter e Configurar Sinalizador de ID de Mensagem de Desabilitação*

### **Interface:**

```
Boolean DisableMessageID  
{  
    get;  
    set;  
}
```

Obter uma indicação de se um aplicativo de recebimento requer que identificadores de mensagens sejam incluídos em mensagens enviadas pelo produtor de mensagens e indicar se um aplicativo de recebimento requer que identificadores de mensagens sejam incluídos em mensagens enviadas pelo produtor de mensagem.

Em uma conexão com um gerenciador de fila ou em uma conexão em tempo real com um broker, esse sinalizador é ignorado. Em uma conexão com um barramento de integração de serviços, o sinalizador é honrado

O ID DisabledMsg possui os valores a seguir:

- `True`, se um aplicativo de recebimento não requerer que identificadores de mensagem sejam incluídos em mensagens enviadas pelo produtor de mensagem
- `False`, se um aplicativo de recebimento exigir que identificadores de mensagem sejam incluídos em mensagens enviadas pelo produtor de mensagens

**Exceções:**

- `XMSEException`

*DisableMsgTS-Obter e Configurar Sinalizador de Registro de Data e Hora de Desativação*

**Interface:**

```
Boolean DisableMessageTimestamp
{
    get;
    set;
}
```

Obter uma indicação se um aplicativo de recebimento requer que registros de data e hora sejam incluídos em mensagens enviadas pelo produtor de mensagens e indicar se um aplicativo de recebimento requer que registros de data e hora sejam incluídos em mensagens enviadas pelo produtor de mensagem.

Em uma conexão em tempo real com um broker, essa sinalização é ignorada. Em uma conexão com um gerenciador de filas ou em uma conexão com um barramento de integração de serviços, a sinalização é honrada.

DisableMsgTS possui os seguintes valores:

- `True`, se um aplicativo de recebimento não precisar que registros de data e hora sejam incluídos em mensagens enviadas pelo produtor de mensagem
- `False`, se um aplicativo de recebimento não precisar que registros de data e hora sejam incluídos em mensagens enviadas pelo produtor de mensagem

**Retorna:**

**Exceções:**

- `XMSEException`

*Prioridade-Obter e Configurar Prioridade Padrão*

**Interface:**

```
Int32 Priority
{
    get;
    set;
}
```

Obter e configurar a prioridade padrão para mensagens enviadas pelo produtor de mensagem

O valor da prioridade da mensagem padrão é um número inteiro no intervalo 0, a prioridade mais baixa, para 9, a prioridade mais alta.

Em uma conexão em tempo real com um broker, a prioridade de uma mensagem é ignorada



**Exceções:**

- XMSEException

*TimeToLive-Obter e configurar o tempo de vida padrão*

**Interface:**

```
Int64 TimeToLive
{
    get;
    set;
}
```

Obter e configurar o período de tempo padrão que uma mensagem existe antes de expirar.

O tempo é medido a partir de quando o produtor da mensagem envia a mensagem e é o tempo padrão de vida em milissegundos Um valor 0 significa que uma mensagem nunca expira.

Para uma conexão em tempo real com um broker, este valor é sempre 0.

**Exceções:**

- XMSEException

**Métodos**

*Fechar-Fechar Produtor de Mensagem*

**Interface:**

```
void Close();
```

Feche o produtor da mensagem.

Se um aplicativo tentar fechar um produtor de mensagem que já está fechado, a chamada será ignorada

**Parâmetros:**

Nenhum

**Retorna:**

Nulo

**Exceções:**

- XMSEException

*Enviar-Enviar*

**Interface:**

```
void Send(IMessage msg) ;
```

Enviar uma mensagem para o destino especificado quando o produtor da mensagem foi criado. Envie a mensagem usando o modo de entrega padrão, prioridade e tempo de vida do produtor da mensagem.

**Parâmetros:****msg (entrada)**

O objeto de Mensagem

**Retorna:**

Nulo

**Exceções:**

- XMSEException

- MessageFormatException
- Exceção de InvalidDestination

*Enviar-Enviar (especificando um modo de entrega, prioridade e tempo de vida)*

**Interface:**

```
void Send(IMessage msg,
         DeliveryMode deliveryMode,
         Int32 priority,
         Int64 timeToLive);
```

Enviar uma mensagem para o destino especificado quando o produtor da mensagem foi criado. Envie a mensagem usando o modo de entrega, prioridade e tempo de vida especificados.

**Parâmetros:**

**msg (entrada)**

O objeto de Mensagem

**deliveryMode (entrada)**

O modo de entrega para a mensagem, que deve ser um dos seguintes valores:

- DeliveryMode.Persistent
- DeliveryMode.NonPersistent

Para uma conexão em tempo real com um broker, o valor deve ser DeliveryMode.NonPersistent.

**prioridade (entrada)**

A prioridade da mensagem. O valor pode ser um número inteiro no intervalo 0, para a prioridade mais baixo, para 9, para a prioridade mais alta. Em uma conexão em tempo real com um broker, o valor é ignorado

**timeToAtivo (entrada)**

O tempo de vida da mensagem em milissegundos. Um valor 0 significa que a mensagem nunca expira. Para uma conexão em tempo real com um broker, o valor deve ser 0

**Retorna:**

Nulo

**Exceções:**

- XMSEException
- MessageFormatException
- Exceção de InvalidDestination
- Exceção IllegalState

*Enviar-Enviar (para um destino especificado)*

**Interface:**

```
void Send(IDestination dest, IMessage msg) ;
```

Enviar uma mensagem para um destino especificado se estiver usando um produtor de mensagem para o qual nenhum destino foi especificado quando o produtor de mensagem foi criado. Envie a mensagem usando o modo de entrega padrão, prioridade e tempo de vida do produtor da mensagem.

Geralmente, você especifica um destino ao criar um produtor de mensagens, mas, se não o fizer, deverá especificar um destino toda vez que enviar uma mensagem.

**Parâmetros:**

**dest (entrada)**

O objeto de Destino

**msg (entrada)**

O objeto de Mensagem

**Retorna:**

Nulo

**Exceções:**

- XMSEException
- MessageFormatException
- Exceção de InvalidDestination

*Enviar-Enviar (para um destino especificado, especificando um modo de entrega, prioridade e tempo de vida)*

**Interface:**

```
void Send(IDestination dest,
          IMessage msg,
          DeliveryMode deliveryMode,
          Int32 priority,
          Int64 timeToLive) ;
```

Enviar uma mensagem para um destino especificado se estiver usando um produtor de mensagem para o qual nenhum destino foi especificado quando o produtor de mensagem foi criado. Envie a mensagem usando o modo de entrega, prioridade e tempo de vida especificados.

Geralmente, você especifica um destino ao criar um produtor de mensagens, mas, se não o fizer, deverá especificar um destino toda vez que enviar uma mensagem.

**Parâmetros:****dest (entrada)**

O objeto de Destino

**msg (entrada)**

O objeto de Mensagem

**deliveryMode (entrada)**

O modo de entrega para a mensagem, que deve ser um dos seguintes valores:

DeliveryMode.Persistent  
DeliveryMode.NonPersistent

Para uma conexão em tempo real com um broker, o valor deve ser DeliveryMode.NonPersistent.

**prioridade (entrada)**

A prioridade da mensagem. O valor pode ser um número inteiro no intervalo 0, para a prioridade mais baixo, para 9, para a prioridade mais alta. Em uma conexão em tempo real com um broker, o valor é ignorado

**timeToAtivo (entrada)**

O tempo de vida da mensagem em milissegundos. Um valor 0 significa que a mensagem nunca expira. Para uma conexão em tempo real com um broker, o valor deve ser 0

**Retorna:**

Nulo

**Exceções:**

- XMSEException
- MessageFormatException
- Exceção de InvalidDestination
- Exceção IllegalState

## Propriedades e métodos herdados

Os métodos a seguir são herdados da interface [IPropertyContext](#):

[GetBooleanProperty](#), [GetByteProperty](#), [GetBytesProperty](#), [GetCharProperty](#), [GetDoubleProperty](#), [GetFloatProperty](#), [GetIntProperty](#), [GetLongProperty](#), [GetObjectProperty](#), [GetShortProperty](#), [GetStringProperty](#), [SetBooleanProperty](#), [SetByteProperty](#), [SetBytesProperty](#), [SetCharProperty](#), [SetDoubleProperty](#), [SetFloatProperty](#), [SetIntProperty](#), [SetLongProperty](#), [SetObjectProperty](#), [SetShortProperty](#), [SetStringProperty](#)

## IOBJECTMESSAGE

Uma mensagem de objeto é uma mensagem cujo corpo compreende um objeto Java ou .NET serializado.

### Hierarquia de herança:

```
IBM.XMS.IPropertyContext
|
+----IBM.XMS.IMessage
|
+----IBM.XMS.IObjectMessage
```

## .NET propriedades

*Objeto-Obter e Configurar Objeto como Bytes.*

### Interface:

```
System.Object Object
{
    get;
    set;
}

Byte[] GetObject();
```

Get e configure o objeto que forma o corpo da mensagem do objeto.

### Exceções:

- [XMSException](#)
- [MessageNotReadableException](#)
- [MessageEOFException](#)
- [MessageNotWritableException](#)

## Propriedades e métodos herdados

As propriedades a seguir são herdadas da interface [IMessage](#):

[JMSCorrelationID](#), [JMSDeliveryMode](#), [JMSDestination](#), [JMSExpiration](#), [JMSPriority](#), [JMSReplyTo](#), [JMSTimestamp](#), [JMSType](#), [Propriedades](#)

Os métodos a seguir são herdados da interface [IMessage](#):

[clearBody](#), [clearProperties](#), [PropertyExists](#)

Os métodos a seguir são herdados da interface [IPropertyContext](#):

[GetBooleanProperty](#), [GetByteProperty](#), [GetBytesProperty](#), [GetCharProperty](#), [GetDoubleProperty](#), [GetFloatProperty](#), [GetIntProperty](#), [GetLongProperty](#), [GetObjectProperty](#), [GetShortProperty](#), [GetStringProperty](#), [SetBooleanProperty](#), [SetByteProperty](#), [SetBytesProperty](#), [SetCharProperty](#), [SetDoubleProperty](#), [SetFloatProperty](#), [SetIntProperty](#), [SetLongProperty](#), [SetObjectProperty](#), [SetShortProperty](#), [SetStringProperty](#)

## IPropertyContext

IPropertyContext é uma superclasse abstrata que contém métodos que obtêm e configuram propriedades. Estes métodos são herdados por outras classes.

### Hierarquia de herança:

Nenhum

### Métodos

*Propriedade GetBoolean-Obter Propriedade Booleana*

#### Interface:

```
Boolean GetBooleanProperty(String property_name);
```

Obter o valor da propriedade booleana com o nome especificado

#### Parâmetros:

**property\_name (entrada)**

Um objeto String encapsulando o nome da propriedade.

#### Retorna:

O valor da propriedade.

#### Contexto do encadeamento:

Determinado pela subclasse

#### Exceções:

- XMSEException

*Propriedade GetByte-Obter Propriedade de Byte*

#### Interface:

```
Byte GetByteProperty(String property_name) ;  
Int16 GetSignedByteProperty(String property_name) ;
```

Obter o valor da propriedade de byte identificada por nome.

#### Parâmetros:

**property\_name (entrada)**

Um objeto String encapsulando o nome da propriedade.

#### Retorna:

O valor da propriedade.

#### Contexto do encadeamento:

Determinado pela subclasse

#### Exceções:

- XMSEException

*Propriedade GetBytes-Propriedade da Matriz de Bytes de Obtenção*

#### Interface:

```
Byte[] GetBytesProperty(String property_name) ;
```

Obtenha o valor da propriedade de matriz de bytes identificada por nome..

**Parâmetros:****property\_name (entrada)**

Um objeto String encapsulando o nome da propriedade.

**Retorna:**

O número de bytes na matriz

**Contexto do encadeamento:**

Determinado pela subclasse

**Exceções:**

- XMSEException

*Propriedade GetChar-Propriedade de caractere de obtenção*

**Interface:**

```
Char GetCharProperty(String property_name) ;
```

Obter o valor da propriedade de caractere de 2 bytes identificada por nome.

**Parâmetros:****property\_name (entrada)**

Um objeto String encapsulando o nome da propriedade.

**Retorna:**

O valor da propriedade.

**Contexto do encadeamento:**

Determinado pela subclasse

**Exceções:**

- XMSEException

*Propriedade GetDouble-Obter Propriedade de Ponto Flutuante de Precisão Dupla*

**Interface:**

```
Double GetDoubleProperty(String property_name) ;
```

Obtenha o valor da propriedade de ponto flutuante de precisão dupla identificada por nome...

**Parâmetros:****property\_name (entrada)**

Um objeto String encapsulando o nome da propriedade.

**Retorna:**

O valor da propriedade.

**Contexto do encadeamento:**

Determinado pela subclasse

**Exceções:**

- XMSEException

*Propriedade GetFloat-Obter propriedade de ponto flutuante*

**Interface:**

```
Single GetFloatProperty(String property_name) ;
```

Obter o valor da propriedade de ponto flutuante identificada por nome.

**Parâmetros:****property\_name (entrada)**

Um objeto String encapsulando o nome da propriedade.

**Retorna:**

O valor da propriedade.

**Contexto do encadeamento:**

Determinado pela subclasse

**Exceções:**

- XMSEException

*Propriedade GetInt-Propriedade GetInt*

**Interface:**

```
Int32 GetIntProperty(String property_name) ;
```

Obter o valor da propriedade de número inteiro identificada pelo nome

**Parâmetros:****property\_name (entrada)**

Um objeto String encapsulando o nome da propriedade.

**Retorna:**

O valor da propriedade.

**Contexto do encadeamento:**

Determinado pela subclasse

**Exceções:**

- XMSEException

*Propriedade GetLong-Obter propriedade de número inteiro longo*

**Interface:**

```
Int64 GetLongProperty(String property_name) ;
```

Obter o valor da propriedade de número inteiro longo identificado por nome.

**Parâmetros:****property\_name (entrada)**

Um objeto String encapsulando o nome da propriedade.

**Retorna:**

O valor da propriedade.

**Contexto do encadeamento:**

Determinado pela subclasse

**Exceções:**

- XMSEException

*Propriedade GetObject-Obter Propriedade do Objeto*

**Interface:**

```
Object GetObjectProperty( String property_name) ;
```

Obtenha o valor e o tipo de dados da propriedade identificados por nome..

**Parâmetros:****property\_name (entrada)**

Um objeto String encapsulando o nome da propriedade.

**Retorna:**

O valor da propriedade, que é um dos seguintes tipos de objeto:

Boolean  
Byte  
Byte[]  
Char  
Double  
Single  
Int32  
Int64  
Int16  
String

**Contexto do encadeamento:**

Determinado pela subclasse

**Exceções:**

- XMSEException

*Propriedade GetShort-Obter propriedade de número inteiro curto*

**Interface:**

```
Int16 GetShortProperty(String property_name) ;
```

Obter o valor da propriedade de número inteiro curto identificada por nome.

**Parâmetros:****property\_name (entrada)**

Um objeto String encapsulando o nome da propriedade.

**Retorna:**

O valor da propriedade.

**Contexto do encadeamento:**

Determinado pela subclasse

**Exceções:**

- XMSEException

*Propriedade GetString-Propriedade GetString*

**Interface:**

```
String GetStringProperty(String property_name) ;
```

Obter o valor da propriedade de sequência identificada por nome.

**Parâmetros:****property\_name (entrada)**

Um objeto String encapsulando o nome da propriedade.

**Retorna:**

Um objeto String encapsulando a sequência que é o valor da propriedade. Se a conversão de dados for necessária, esse valor será a cadeia após a conversão..



**Contexto do encadeamento:**

Determinado pela subclasse

**Exceções:**

- XMSException

*Propriedade SetBoolean-Configurar Propriedade Booleana*

**Interface:**

```
void SetBooleanProperty( String property_name, Boolean value) ;
```

Configure o valor da propriedade booleana identificada por nome

**Parâmetros:****property\_name (entrada)**

Um objeto String encapsulando o nome da propriedade.

**valor (entrada)**

O valor da propriedade.

**Retorna:**

Nulo

**Contexto do encadeamento:**

Determinado pela subclasse

**Exceções:**

- XMSException
- MessageNotWritableException

*Propriedade SetByte-Configurar propriedade de byte*

**Interface:**

```
void SetByteProperty( String property_name, Byte value) ;  
void SetSignedByteProperty( String property_name, Int16 value) ;
```

Configure o valor da propriedade de byte identificada por nome..

**Parâmetros:****property\_name (entrada)**

Um objeto String encapsulando o nome da propriedade.

**valor (entrada)**

O valor da propriedade.

**Retorna:**

Nulo

**Contexto do encadeamento:**

Determinado pela subclasse

**Exceções:**

- XMSException
- MessageNotWritableException

*Propriedade SetBytes-Configurar Propriedade da Matriz de Bytes*

**Interface:**

```
void SetBytesProperty( String property_name, Byte[] value ) ;
```

Configure o valor da propriedade de matriz de bytes identificada por nome..

**Parâmetros:**

**property\_name (entrada)**

Um objeto String encapsulando o nome da propriedade.

**valor (entrada)**

O valor da propriedade, que é uma matriz de bytes.

**Retorna:**

Nulo

**Contexto do encadeamento:**

Determinado pela subclasse

**Exceções:**

- XMSEException
- MessageNotWritableException

*Propriedade SetChar-Configurar Propriedade de Caractere*

**Interface:**

```
void SetCharProperty( String property_name, Char value) ;
```

Configure o valor da propriedade de caractere de 2 bytes identificada por nome..

**Parâmetros:**

**property\_name (entrada)**

Um objeto String encapsulando o nome da propriedade.

**valor (entrada)**

O valor da propriedade.

**Retorna:**

Nulo

**Contexto do encadeamento:**

Determinado pela subclasse

**Exceções:**

- XMSEException
- MessageNotWritableException

*Propriedade SetDouble-Configurar Propriedade de Ponto Flutuante de Precisão Dupla*

**Interface:**

```
void SetDoubleProperty( String property_name, Double value) ;
```

Configure o valor da propriedade de ponto flutuante de precisão dupla identificado por nome..

**Parâmetros:**

**property\_name (entrada)**

Um objeto String encapsulando o nome da propriedade.

**valor (entrada)**

O valor da propriedade.

**Retorna:**

Nulo

**Contexto do encadeamento:**

Determinado pela subclasse

**Exceções:**

- XMSEException
- MessageNotWritableException

*Propriedade SetFloat-Configurar propriedade de ponto flutuante*

**Interface:**

```
void SetFloatProperty( String property_name, Single value) ;
```

Configure o valor da propriedade de ponto flutuante identificada por nome..

**Parâmetros:****property\_name (entrada)**

Um objeto String encapsulando o nome da propriedade.

**valor (entrada)**

O valor da propriedade.

**Retorna:**

Nulo

**Contexto do encadeamento:**

Determinado pela subclasse

**Exceções:**

- XMSEException
- MessageNotWritableException

*Propriedade SetInt-Configurar propriedade de número inteiro*

**Interface:**

```
void SetIntProperty( String property_name, Int32 value) ;
```

Configure o valor da propriedade de número inteiro identificada por nome.

**Parâmetros:****property\_name (entrada)**

Um objeto String encapsulando o nome da propriedade.

**valor (entrada)**

O valor da propriedade.

**Retorna:**

Nulo

**Contexto do encadeamento:**

Determinado pela subclasse

**Exceções:**

- XMSEException
- MessageNotWritableException

*Propriedade SetLong-Configurar propriedade de número inteiro longo*

**Interface:**

```
void SetLongProperty( String property_name, Int64 value) ;
```

Configurar o valor da propriedade de número inteiro longo identificada pelo nome.

**Parâmetros:****property\_name (entrada)**

Um objeto String encapsulando o nome da propriedade.

**valor (entrada)**

O valor da propriedade.

**Retorna:**

Nulo

**Contexto do encadeamento:**

Determinado pela subclasse

**Exceções:**

- XMSEException
- MessageNotWritableException

*Propriedade SetObject-Configurar Propriedade do objeto*

**Interface:**

```
void SetObjectProperty( String property_name, Object value) ;
```

Configure o valor e o tipo de dado de uma propriedade identificada por nome.

**Parâmetros:****property\_name (entrada)**

Um objeto String encapsulando o nome da propriedade.

**objectType (entrada)**

O valor da propriedade, que deve ser um dos seguintes tipos de objeto:

Boolean  
Byte  
Byte[]  
Char  
Double  
Single  
Int32  
Int64  
Int16  
String

**valor (entrada)**

O valor da propriedade como uma matriz de bytes.

**comprimento (entrada)**

O número de bytes na matriz

**Retorna:**

Nulo

**Contexto do encadeamento:**

Determinado pela subclasse

**Exceções:**

- XMSEException
- MessageNotWritableException

*Propriedade SetShort-Configurar propriedade de número inteiro curto*

**Interface:**

```
void SetShortProperty( String property_name, Int16 value) ;
```

Configure o valor da propriedade de número inteiro curto identificada por nome..

**Parâmetros:**

**property\_name (entrada)**

Um objeto String encapsulando o nome da propriedade.

**valor (entrada)**

O valor da propriedade.

**Retorna:**

Nulo

**Contexto do encadeamento:**

Determinado pela subclasse

**Exceções:**

- XMSEException
- MessageNotWritableException

*Propriedade SetString-Configurar Propriedade de Cadeia*

**Interface:**

```
void SetStringProperty( String property_name, String value);
```

Configure o valor da propriedade de cadeia identificada por nome..

**Parâmetros:**

**property\_name (entrada)**

Um objeto String encapsulando o nome da propriedade.

**valor (entrada)**

Um objeto String encapsulando a sequência que é o valor da propriedade.

**Retorna:**

Nulo

**Contexto do encadeamento:**

Determinado pela subclasse

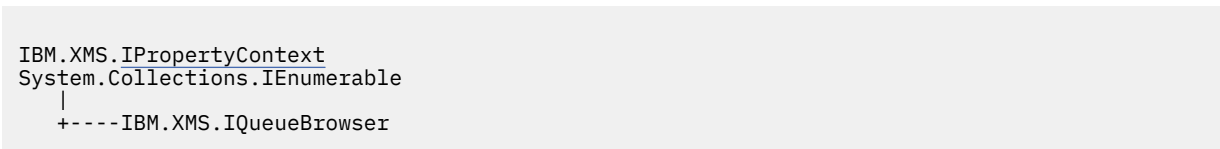
**Exceções:**

- XMSEException
- MessageNotWritableException

## **IQueueBrowser**

Um aplicativo usa um navegador de filas para pesquisar mensagens em uma fila sem removê-las.

**Hierarquia de herança:**



## **.NET propriedades**

## *MessageSelector -Obter Seletor de Mensagem*

### **Interface:**

```
String MessageSelector
{
    get;
}
```

Obtenha o seletor de mensagem do navegador de fila.

O seletor de mensagem é um objeto String que encapsula a expressão do seletor de mensagem. Se a conversão de dados for necessária, esse valor será a expressão do seletor de mensagem após a conversão.. Se o navegador da fila não tiver um seletor de mensagens, o método retornará um objeto String nulo.

### **Exceções:**

- XMSEException

## *Fila-Fila de obtenção*

### **Interface:**

```
IDestination Queue
{
    get;
}
```

Obtenha a fila associada ao navegador de fila como um objeto de destino que representa a fila

### **Exceções:**

- XMSEException

## **Métodos**

## *Fechar-Fechar Navegador da Fila*

### **Interface:**

```
void Close();
```

Feche o navegador da filas.

Se um aplicativo tentar fechar um browser de fila que já esteja fechado, a chamada será ignorada

### **Parâmetros:**

Nenhum

### **Retorna:**

Nulo

### **Exceções:**

- XMSEException

## *GetEnumerator -Obter mensagens*

### **Interface:**

```
IEnumerator GetEnumerator();
```

Obter uma lista das mensagens na fila

O método retorna um enumerador que encapsula uma lista de objetos de Mensagem. A ordem dos objetos de Mensagem é igual à ordem na qual as mensagens seriam recuperadas da fila. O aplicativo pode então usar o enumerador para pesquisar cada mensagem por vez.

O enumerador é atualizado dinamicamente conforme as mensagens são colocadas na fila e removidas da fila. Toda vez que o aplicativo chama `IEnumerator.MoveNext ()` para procurar a próxima mensagem na fila, a mensagem reflete o conteúdo atual da fila.

Se um aplicativo chamar esse método mais de uma vez para um navegador de filas, cada chamada retornará um novo enumerador. O aplicativo pode, portanto, usar mais de um enumerador para procurar as mensagens em uma fila e manter várias posições dentro da fila.

**Parâmetros:**

Nenhum

**Retorna:**

O objeto do Iterator (Iterator)

**Exceções:**

- `XMSEException`

***Propriedades e métodos herdados***

Os métodos a seguir são herdados da interface [IPropertyContext](#):

[GetBooleanProperty](#), [GetByteProperty](#), [GetBytesProperty](#), [GetCharProperty](#), [GetDoubleProperty](#), [GetFloatProperty](#), [GetIntProperty](#), [GetLongProperty](#), [GetObjectProperty](#), [GetShortProperty](#), [GetStringProperty](#), [SetBooleanProperty](#), [SetByteProperty](#), [SetBytesProperty](#), [SetCharProperty](#), [SetDoubleProperty](#), [SetFloatProperty](#), [SetIntProperty](#), [SetLongProperty](#), [SetObjectProperty](#), [SetShortProperty](#), [SetStringProperty](#)

**Solicitante**

Um aplicativo usa um solicitante para enviar uma mensagem de solicitação e, em seguida, aguardar e receber a resposta..

**Hierarquia de herança:**

Nenhum

**Construtores**

*Solicitante-Criar Solicitante*

**Interface:**

```
Requestor(ISession sess, IDestination dest);
```

Criar um solicitante.

**Parâmetros:**

**sess (entrada)**

Um objeto de Sessão. A sessão não deve ser transacionada e deve ter um dos seguintes modos de confirmação:

- `AcknowledgeMode.AutoAcknowledge`
- `AcknowledgeMode.DupsOkAcknowledge`

**dest (entrada)**

Um objeto de Destino representando o destino no qual o aplicativo pode enviar mensagens de solicitação.

**Contexto do encadeamento:**

A sessão associada ao solicitante

**Exceções:**

- XMSEException

**Métodos***Fechar-Fechar Solicitante***Interface:**

```
void Close();
```

Feche o solicitante.

Se um aplicativo tentar fechar um solicitante que já esteja fechado, a chamada será ignorada..

**Nota:** Quando um aplicativo fecha um solicitante, a sessão associada não é fechada também Nesse aspecto, XMS se comporta de forma diferente em comparação com JMS.

**Parâmetros:**

Nenhum

**Retorna:**

Nulo

**Contexto do encadeamento:**

Qualquer

**Exceções:**

- XMSEException

*Solicitação-Resposta de solicitação***Interface:**

```
IMessage Request(IMessage requestMessage);
```

Envie uma mensagem de solicitação e, em seguida, aguarde e receba uma resposta do aplicativo que recebe a mensagem de solicitação

Uma chamada para esse método é bloqueada até que uma resposta seja recebida ou até que a sessão termine, o que ocorrer primeiro.

**Parâmetros:****requestMessage (entrada)**

O objeto de Mensagem encapsulando a mensagem de solicitação

**Retorna:**

Um ponteiro para o objeto de Mensagem que contém a mensagem de resposta.

**Contexto do encadeamento:**

A sessão associada ao solicitante

**Exceções:**

- XMSEException

**Exceção de ResourceAllocation**

XMS lança essa exceção se XMS não puder alocar os recursos necessários por um método.

**Hierarquia de herança:**

```
IBM.XMS.XMSEException
```



```
+----IBM.XMS.XMSEnception
      |
      +----IBM.XMS.ResourceAllocationException
```

### **Propriedades e métodos herdados**

Os métodos a seguir são herdados da interface [XMSEnception](#):

[GetErrorCódigo](#), [GetLinkedExceção](#)

## **SecurityException**

O XMS lança essa exceção se o identificador de usuário e a senha fornecidos para autenticar um aplicativo forem rejeitados XMS também lança essa exceção se uma verificação de autoridade falhar e impedir que um método seja concluído.

### **Hierarquia de herança:**

```
IBM.XMS.XMSEnception
|
+----IBM.XMS.XMSEnception
      |
      +----IBM.XMS.SecurityException
```

### **Propriedades e métodos herdados**

Os métodos a seguir são herdados da interface [XMSEnception](#):

[GetErrorCódigo](#), [GetLinkedExceção](#)

## **ISession**

Uma sessão é um único contexto encadeado para enviar e receber mensagens.

### **Hierarquia de herança:**

```
IBM.XMS.IPropertyContext
|
+----IBM.XMS.ISession
```

Para obter uma lista das propriedades definidas XMS de um objeto Session, consulte [“Propriedades da Sessão.”](#) na página 2114.

## **.NET propriedades**

*AcknowledgeMode -Obter Modo de Confirmação*

### **Interface:**

```
AcknowledgeMode AcknowledgeMode
{
    get;
}
```

Obter o modo de confirmação da sessão.

O modo de confirmação é especificado quando a sessão é criada

Desde que a sessão não seja transacionada, o modo de reconhecimento será um dos seguintes valores:

```
AcknowledgeMode.AutoAcknowledge
AcknowledgeMode.ClientAcknowledge
AcknowledgeMode.DupsOkAcknowledge
```

Para obter mais informações sobre os modos de confirmação, consulte [Confirmação de mensagem](#)

Uma sessão transacionada não possui modo de confirmação. Se a sessão for transacionada, o método retornará `AcknowledgeMode.SessionTransacted` no lugar.

**Exceções:**

- `XMSEException`

*Transacionado-Determine se Transacionado*

**Interface:**

```
Boolean Transacted
{
    get;
}
```

Determine se a sessão foi transacionada

O transacionado declarado é:

- `True`, se a sessão for transacionada..
- `False`, se a sessão não for transacionada..

Para uma conexão em tempo real com um broker, o método sempre retorna `False`..

**Exceções:**

- `XMSEException`

**Métodos**

*Fechar-Fechar Sessão*

**Interface:**

```
void Close();
```

Feche a sessão. Se a sessão for transacionada, qualquer transação em andamento será retrocedida

Se um aplicativo tentar fechar uma sessão que já tenha sido fechada, a chamada será ignorada

**Parâmetros:**

Nenhum

**Retorna:**

Nulo

**Contexto do encadeamento:**

Qualquer

**Exceções:**

- `XMSEException`

*Confirmação-Confirmar*

**Interface:**

```
void Commit();
```

Confirmar todas as mensagens processada na transação atual

A sessão deve ser uma sessão transacionada

**Parâmetros:**

Nenhum

**Retorna:**

Nulo

**Exceções:**

- XMSEException
- Exceção IllegalStateException
- TransactionRolledBackException

*CreateBrowser -Criar Navegador da Fila*

**Interface:**

```
IQueueBrowser CreateBrowser(IDestination queue) ;
```

Crie um navegador de filas da fila especificada.

**Parâmetros:****fila (entrada)**

Um objeto de Destino que representa a fila

**Retorna:**

O objeto QueueBrowser .

**Exceções:**

- XMSEException
- Exceção de InvalidDestination

*CreateBrowser -Create Queue Browser (com seletor de mensagens)*

**Interface:**

```
IQueueBrowser CreateBrowser(IDestination queue, String selector) ;
```

Crie um navegador de fila para a fila especificada usando um seletor de mensagem

**Parâmetros:****fila (entrada)**

Um objeto de Destino que representa a fila

**seletor (entrada)**

Um objeto String encapsulando uma expressão do seletor de mensagem. Apenas as mensagens com propriedades que correspondem à expressão do seletor de mensagem são entregues para o navegador de filas

Um objeto String nulo significa que não há nenhum seletor de mensagens para o navegador de fila

**Retorna:**

O objeto QueueBrowser .

**Exceções:**

- XMSEException
- Exceção de InvalidDestination
- InvalidSelectorExceção

## *Mensagem CreateBytes-Criar Mensagem de Bytes*

### **Interface:**

```
IBytesMessage CreateBytesMessage();
```

Crie uma mensagem de bytes

### **Parâmetros:**

Nenhum

### **Retorna:**

O objeto BytesMessage ..

### **Exceções:**

- XMSEException
- IllegalStateExceção (A sessão foi encerrada)

## *CreateConsumer -Criar Consumidor*

### **Interface:**

```
IMessageConsumer CreateConsumer(IDestination dest) ;
```

Crie um consumidor de mensagens para o destino especificado

### **Parâmetros:**

#### **dest (entrada)**

O objeto de Destino

### **Retorna:**

O objeto MessageConsumer ..

### **Exceções:**

- XMSEException
- Exceção de InvalidDestination

## *CreateConsumer -Criar Consumidor (com seletor de mensagens).*

### **Interface:**

```
IMessageConsumer CreateConsumer(IDestination dest,  
                                String selector) ;
```

Criar um consumidor de mensagens para o destino especificado usando um seletor de mensagens

### **Parâmetros:**

#### **dest (entrada)**

O objeto de Destino

#### **seletor (entrada)**

Um objeto String encapsulando uma expressão do seletor de mensagem. Apenas as mensagens com propriedades que correspondem à expressão do seletor de mensagem são entregues para o consumidor de mensagem

Um objeto String nulo significa que não há seletor de mensagem para o consumidor de mensagens.

### **Retorna:**

O objeto MessageConsumer ..

**Exceções:**

- XMSEException
- Exceção de InvalidDestination
- InvalidSelectorExceção

*CreateConsumer - Criar Consumidor (com seletor de mensagens e sinalização de mensagem local)*

**Interface:**

```
IMessageConsumer CreateConsumer(IDestination dest,  
                                String selector,  
                                Boolean noLocal) ;
```

Crie um consumidor de mensagens para o destino especificado utilizando um seletor de mensagens e, se o destino for um tópico, especificando se o consumidor de mensagens recebe as mensagens publicadas pela sua própria conexão

**Parâmetros:****dest (entrada)**

O objeto de Destino

**seletor (entrada)**

Um objeto String encapsulando uma expressão do seletor de mensagem. Apenas as mensagens com propriedades que correspondem à expressão do seletor de mensagem são entregues para o consumidor de mensagem

Um objeto String nulo significa que não há seletor de mensagem para o consumidor de mensagens.

**noLocal (entrada)**

O valor True significa que o consumidor de mensagens não recebe as mensagens publicadas por sua própria conexão O valor False significa que o consumidor de mensagens não recebe as mensagens publicadas por sua própria conexão. O valor padrão é False.

**Retorna:**

O objeto MessageConsumer ..

**Exceções:**

- XMSEException
- Exceção de InvalidDestination
- InvalidSelectorExceção

*CreateDurableAssinante-Criar Assinante Durável*

**Interface:**

```
IMessageConsumer CreateDurableSubscriber(IDestination dest,  
                                          String subscription) ;
```

Crie um assinante durável para o tópico especificado

Este método não é válido para uma conexão em tempo real com um broker

Para obter mais informações sobre assinantes duráveis, consulte [Assinantes duráveis](#).

**Parâmetros:****dest (entrada)**

Um objeto de Destino que representa o tópico O tópico não deve ser temporário.

**assinatura (entrada)**

Um objeto String encapsulando um nome que identifica a assinatura durável. O nome deve ser exclusivo no identificador de cliente para a conexão.

**Retorna:**

O objeto MessageConsumer representando o assinante durável.

**Exceções:**

- XMSException
- Exceção de InvalidDestination

*CreateDurableAssinante-Criar Assinante Durável (com seletor de mensagem e sinalizador de mensagem local)*

**Interface:**

```
IMessageConsumer CreateDurableSubscriber(IDestination dest,  
                                         String subscription,  
                                         String selector,  
                                         Boolean noLocal) ;
```

Crie um assinante durável para o tópico especificado usando um seletor de mensagem e especificando se o assinante durável recebe as mensagens publicadas pela sua própria conexão

Este método não é válido para uma conexão em tempo real com um broker

Para obter mais informações sobre assinantes duráveis, consulte [Assinantes duráveis](#).

**Parâmetros:****dest (entrada)**

Um objeto de Destino que representa o tópico O tópico não deve ser temporário.

**assinatura (entrada)**

Um objeto String encapsulando um nome que identifica a assinatura durável. O nome deve ser exclusivo no identificador de cliente para a conexão.

**seletor (entrada)**

Um objeto String encapsulando uma expressão do seletor de mensagem. Apenas as mensagens com propriedades que correspondem à expressão do seletor de mensagem são entregues para o assinante durável

Um objeto String nulo significa que não há seletor de mensagem para o assinante durável.

**noLocal (entrada)**

O valor True significa que o assinante durável não recebe as mensagens publicadas por sua própria conexão O valor False significa que o assinante durável não recebe as mensagens publicadas por sua própria conexão. O valor padrão é False.

**Retorna:**

O objeto MessageConsumer representando o assinante durável.

**Exceções:**

- XMSException
- Exceção de InvalidDestination
- InvalidSelectorExceção

*Mensagem CreateMap-Criar Mensagem de Mapa*

**Interface:**

```
IMapMessage CreateMapMessage();
```

Crie uma mensagem de mapa.

**Parâmetros:**

Nenhum

**Retorna:**

O objeto MapMessage ..

**Exceções:**

- XMSException
- IllegalStateException (A sessão foi encerrada)

*CreateMessage -Criar mensagem*

**Interface:**

```
IMessage CreateMessage();
```

Crie uma mensagem que não tenha corpo.

**Parâmetros:**

Nenhum

**Retorna:**

O objeto de Mensagem

**Exceções:**

- XMSException
- IllegalStateException (A sessão foi encerrada)

*Mensagem CreateObject-Criar Mensagem de Objeto*

**Interface:**

```
IObjectMessage CreateObjectMessage();
```

Criar uma mensagem de objeto

**Parâmetros:**

Nenhum

**Retorna:**

O objeto ObjectMessage ..

**Exceções:**

- XMSException
- IllegalStateException (A sessão foi encerrada)

*CreateProducer -Criar Produtor*

**Interface:**

```
IMessageProducer CreateProducer(IDestination dest) ;
```

Crie um produtor de mensagem para enviar mensagens ao destino especificado.

**Parâmetros:****dest (entrada)**

O objeto de Destino

Se você especificar um objeto de Destino nulo, o produtor de mensagem será criado sem um destino Nesse caso, o aplicativo deve especificar um destino sempre que usar o produtor de mensagem para enviar uma mensagem.

**Retorna:**

O objeto MessageProducer .

**Exceções:**

- XMSEException
- Exceção de InvalidDestination

*CreateQueue -Criar Fila*

**Interface:**

```
IDestination CreateQueue(String queue) ;
```

Crie um objeto de Destino para representar uma fila no servidor de sistema de mensagens

Este método não cria a fila no servidor de mensagens. Deve-se criar a fila antes que um aplicativo possa chamar esse método

**Parâmetros:****fila (entrada)**

Um objeto String encapsulando o nome da fila ou encapsulando um identificador uniforme de recursos (URI) que identifica a fila.

**Retorna:**

O objeto de Destino que representa a fila

**Exceções:**

- XMSEException

*Mensagem CreateStream-Criar Mensagem de Fluxo*

**Interface:**

```
IStreamMessage CreateStreamMessage();
```

Criar uma mensagem de fluxo

**Parâmetros:**

Nenhum

**Retorna:**

O objeto StreamMessage ..

**Exceções:**

- XMSEException
- XMS\_ILLEGAL\_STATE\_EXCEPTION

*Fila CreateTemporary-Criar Fila Temporária*

**Interface:**

```
IDestination CreateTemporaryQueue() ;
```

Crie uma fila temporária.

O escopo da fila temporária é a conexão Apenas as sessões criadas pela conexão podem usar a fila temporária

A fila temporária permanece até que seja explicitamente excluída ou a conexão termine, o que for anterior.

Para obter mais informações sobre filas temporárias, consulte [Destinos temporários](#).

**Parâmetros:**

Nenhum



**Retorna:**

O objeto de Destino que representa a fila temporária

**Exceções:**

- XMSEException

*Tópico CreateTemporary-Criar tópico temporário*

**Interface:**

```
IDestination CreateTemporaryTopic() ;
```

Crie um tópico temporário

O escopo do tópico temporário é a conexão. Apenas as sessões criadas pela conexão podem usar o tópico temporário

O tópico temporário permanece até que seja excluído explicitamente ou a conexão seja encerrada, o que ocorrer primeiro.

Para obter mais informações sobre tópicos temporários, consulte [Destinos provisórios](#)

**Parâmetros:**

Nenhum

**Retorna:**

O objeto de Destino que representa o tópico temporário

**Exceções:**

- XMSEException

*Mensagem CreateText-Criar Mensagem de Texto*

**Interface:**

```
ITextMessage CreateTextMessage();
```

Crie uma mensagem de texto com um corpo vazio

**Parâmetros:**

Nenhum

**Retorna:**

O objeto TextMessage ..

**Exceções:**

- XMSEException

*Mensagem CreateText-Criar Mensagem de Texto (inicializada)*

**Interface:**

```
ITextMessage CreateTextMessage(String initialValue);
```

Crie uma mensagem de texto cujo corpo seja inicializado com o texto especificado

**Parâmetros:****initialValue (entrada)**

Um objeto String encapsulando o texto para inicializar o corpo da mensagem de texto.

Nenhum

**Retorna:**

O objeto TextMessage ..

**Exceções:**

- XMSEException

*CreateTopic -Criar Tópico*

**Interface:**

```
IDestination CreateTopic(String topic) ;
```

Crie um objeto de Destino para representar um tópico

**Parâmetros:****tópico (entrada)**

Um objeto String encapsulando o nome do tópico ou encapsulando um identificador uniforme de recursos (URI) que identifica o tópico.

**Retorna:**

O objeto de Destino que representa o tópico

**Exceções:**

- XMSEException

*Recuperar-Recuperar*

**Interface:**

```
void Recover();
```

Recuperar a sessão. A entrega de mensagem é interrompida e, em seguida, reiniciada com a mensagem não reconhecida mais antiga

A sessão não deve ser uma sessão transacionada

Para obter mais informações sobre a recuperação de uma sessão, consulte [Confirmação de Mensagem](#)

**Parâmetros:**

Nenhum

**Retorna:**

Nulo

**Exceções:**

- XMSEException
- Exceção IllegalStateException

*Retrocesso-Retrocesso*

**Interface:**

```
void Rollback();
```

Retorne todas as mensagens processadas na transação atual.

A sessão deve ser uma sessão transacionada

**Parâmetros:**

Nenhum

**Retorna:**

Nulo

**Exceções:**

- XMSEException

- Exceção `IllegalState`

*Cancelar assinatura-Cancelar assinatura*

**Interface:**

```
void Unsubscribe(String subscription);
```

Excluir uma assinatura durável. O servidor de mensagens exclui o registro da assinatura durável que ele está mantendo e não envia mais mensagens para o assinante durável.

Um aplicativo não pode excluir uma assinatura durável em qualquer uma das seguintes circunstâncias:

- Enquanto há um consumidor de mensagens ativo para a assinatura durável
- Enquanto uma mensagem consumida faz parte de uma transação pendente
- Enquanto uma mensagem consumida não foi reconhecida

Este método não é válido para uma conexão em tempo real com um broker

**Parâmetros:**

**assinatura (entrada)**

Um objeto `String` encapsulando o nome que identifica a assinatura durável.

**Retorna:**

Nulo

**Exceções:**

- `XMSEException`
- Exceção de `InvalidDestination`
- Exceção `IllegalState`

**Propriedades e métodos herdados**

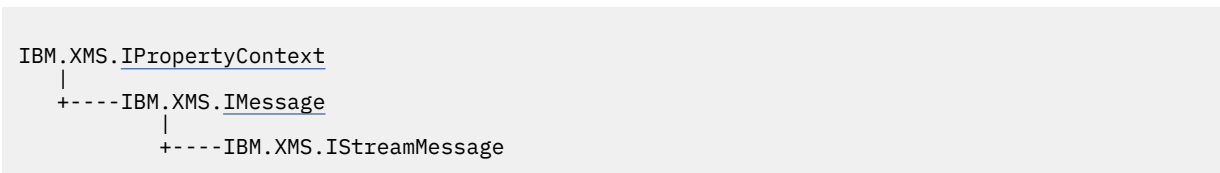
Os métodos a seguir são herdados da interface [IPropertyContext](#):

[GetBooleanProperty](#), [GetByteProperty](#), [GetBytesProperty](#), [GetCharProperty](#), [GetDoubleProperty](#), [GetFloatProperty](#), [GetIntProperty](#), [GetLongProperty](#), [GetObjectProperty](#), [GetShortProperty](#), [GetStringProperty](#), [SetBooleanProperty](#), [SetByteProperty](#), [SetBytesProperty](#), [SetCharProperty](#), [SetDoubleProperty](#), [SetFloatProperty](#), [SetIntProperty](#), [SetLongProperty](#), [SetObjectProperty](#), [SetShortProperty](#), [SetStringProperty](#)

**IStreamMessage**

Uma mensagem de fluxo é uma mensagem cujo corpo compreende um fluxo de valores, em que cada valor possui um tipo de dados associado. Os conteúdos do corpo são gravados e lidos sequencialmente.

**Hierarquia de herança:**



Quando um aplicativo lê um valor do fluxo de mensagens, o valor pode ser convertido pelo XMS em outro tipo de dados. Para obter mais informações sobre essa forma de conversão implícita, consulte [O corpo de uma mensagem XMS](#)

**Métodos**

*ReadBoolean -Valor Booleano de leitura*

**Interface:**

```
Boolean ReadBoolean();
```

Leia um valor booleano do fluxo de mensagens.

**Parâmetros:**

Nenhum

**Retorna:**

O valor booleano que é lido.

**Exceções:**

- XMSEException
- MessageNotReadableException
- MessageEOFException

*ReadByte -Byte de Leitura*

**Interface:**

```
Int16 ReadSignedByte();  
Byte ReadByte();
```

Leia um número inteiro de 8 bits assinado do fluxo de mensagens.

**Parâmetros:**

Nenhum

**Retorna:**

O byte lido.

**Exceções:**

- XMSEException
- MessageNotReadableException
- MessageEOFException

*ReadBytes -Bytes de leitura*

**Interface:**

```
Int32 ReadBytes(Byte[] array);
```

Ler uma matriz de bytes do fluxo de mensagens.

**Parâmetros:**

**matriz (entrada)**

O buffer contendo a matriz de bytes que é lida e o comprimento do buffer em bytes.

Se o número de bytes na matriz for menor ou igual ao comprimento do buffer, a matriz inteira será lida no buffer. Se o número de bytes na matriz for maior que o comprimento do buffer, o buffer será preenchido com parte da matriz e um cursor interno marcará a posição do byte seguinte a ser lido. Uma chamada subsequente para readBytes() lê bytes a partir da matriz iniciando na posição atual do cursor.

Se você especificar um ponteiro nulo na entrada, a chamada ignorará a matriz de bytes sem lê-la.

**Retorna:**

O número de bytes lidos no buffer. Se o buffer for parcialmente preenchido, o valor será menor que o comprimento do buffer, indicando que não há mais bytes na matriz restantes a serem

lidos. Se não houver bytes restantes a serem lidos na matriz antes da chamada, o valor será `XMSC_END_OF_BYTEARRAY`.

Se você especificar um ponteiro nulo na entrada, o método não retornará valor.

**Exceções:**

- `XMSEException`
- `MessageNotReadableException`
- `MessageEOFException`

*ReadChar -Caractere de Leitura*

**Interface:**

```
Char ReadChar();
```

Leia um caractere de 2 bytes do fluxo de mensagens.

**Parâmetros:**

Nenhum

**Retorna:**

O caractere lido.

**Exceções:**

- `XMSEException`
- `MessageNotReadableException`
- `MessageEOFException`

*ReadDouble -Ler Número de Ponto Flutuante de Precisão Dupla*

**Interface:**

```
Double ReadDouble();
```

Leia um número de ponto flutuante de precisão dupla de 8 bytes a partir do fluxo de mensagens

**Parâmetros:**

Nenhum

**Retorna:**

O número do ponto flutuante de precisão dupla que é lido

**Exceções:**

- `XMSEException`
- `MessageNotReadableException`
- `MessageEOFException`

*ReadFloat -Ler número de ponto flutuante*

**Interface:**

```
Single ReadFloat();
```

Leia um número de ponto flutuante de 4 bytes a partir do fluxo de mensagens

**Parâmetros:**

Nenhum

**Retorna:**

O número de ponto flutuante que é lido

**Exceções:**

- XMSEException
- MessageNotReadableException
- MessageEOFException

*ReadInt - Ler Número Inteiro*

**Interface:**

```
Int32 ReadInt();
```

Leia um número inteiro assinado de 32 bits do fluxo de mensagens.

**Parâmetros:**

Nenhum

**Retorna:**

O número inteiro que é lido.

**Exceções:**

- XMSEException
- MessageNotReadableException
- MessageEOFException

*ReadLong - Número inteiro longo de leitura*

**Interface:**

```
Int64 ReadLong();
```

Leia um número inteiro assinado de 64 bits a partir do fluxo de mensagens.

**Parâmetros:**

Nenhum

**Retorna:**

O número inteiro longo que é lido.

**Exceções:**

- XMSEException
- MessageNotReadableException
- MessageEOFException

*ReadObject - Ler Objeto*

**Interface:**

```
Object ReadObject();
```

Leia um valor do fluxo de mensagens e retorne seu tipo de dados.

**Parâmetros:**

Nenhum

**Retorna:**

O valor, que é um dos seguintes tipos de objeto:

Boolean  
Byte  
Byte[]

Char  
Double  
Single  
Int32  
Int64  
Int16  
String

**Exceções:**

XMSEException

*ReadShort -Número Inteiro Curto de Leitura*

**Interface:**

```
Int16 ReadShort();
```

Leia um número inteiro de 16 bits assinado a partir do fluxo de mensagens.

**Parâmetros:**

Nenhum

**Retorna:**

O número inteiro curto que é lido.

**Exceções:**

- XMSEException
- MessageNotReadableException
- MessageEOFException

*ReadString -Sequência de Leitura*

**Interface:**

```
String ReadString();
```

Leia uma cadeia do fluxo de mensagens. Se necessário, XMS converte os caracteres na sequência na página de código local.

**Parâmetros:**

Nenhum

**Retorna:**

Um objeto String encapsulando a sequência que é lida. Se a conversão de dados for necessária, esta será a sequência após a conversão

**Exceções:**

- XMSEException
- MessageNotReadableException
- MessageEOFException

*Reconfigurar-Reconfigurar*

**Interface:**

```
void Reset();
```

Coloque o corpo da mensagem no modo somente leitura e posicione o cursor no início do fluxo de mensagens.

**Parâmetros:**

Nenhum

**Retorna:**

Nulo

**Exceções:**

- XMSEException
- MessageNotReadableException
- MessageEOFException

*WriteBoolean -Gravar Valor Booleano*

**Interface:**

```
void WriteBoolean(Boolean value);
```

Grave um valor booleano no fluxo de mensagens.

**Parâmetros:****valor (entrada)**

O valor booleano a ser gravado.

**Retorna:**

Nulo

**Exceções:**

- XMSEException
- MessageNotWritableException

*WriteByte -Byte de gravação*

**Interface:**

```
void WriteByte(Byte value);  
void WriteSignedByte(Int16 value);
```

Grave um byte no fluxo de mensagens

**Parâmetros:****valor (entrada)**

O byte a ser gravado..

**Retorna:**

Nulo

**Exceções:**

- XMSEException
- MessageNotWritableException

*WriteBytes -Bytes de Gravação*

**Interface:**

```
void WriteBytes(Byte[] value);
```

Gravar uma matriz de bytes no fluxo de mensagens.



**Parâmetros:****valor (entrada)**

A matriz de bytes a ser gravada

**comprimento (entrada)**

O número de bytes na matriz

**Retorna:**

Nulo

**Exceções:**

- XMSEException
- MessageNotWritableException

*WriteChar -Caractere de gravação*

**Interface:**

```
void WriteChar(Char value);
```

Grave um caractere no fluxo de mensagens como 2 bytes, primeiro byte de alta ordem.

**Parâmetros:****valor (entrada)**

O caractere a ser gravado

**Retorna:**

Nulo

**Exceções:**

- XMSEException
- MessageNotWritableException

*WriteDouble -Número do ponto flutuante de precisão dupla de gravação*

**Interface:**

```
void WriteDouble(Double value);
```

Converta um número de vírgula flutuante de precisão dupla em um número inteiro longo e grave o número inteiro longo no fluxo de mensagem como 8 bytes, primeiro byte de alta ordem...

**Parâmetros:****valor (entrada)**

O número de ponto flutuante de precisão dupla a ser gravado

**Retorna:**

Nulo

**Exceções:**

- XMSEException
- MessageNotWritableException

*WriteFloat -Número do ponto flutuante de gravação*

**Interface:**

```
void WriteFloat(Single value);
```

Converta um número de ponto flutuante em um número inteiro e escreva o número inteiro no fluxo de mensagens como 4 bytes, primeiro byte de alta ordem.

**Parâmetros:**

**valor (entrada)**

O número de ponto flutuante a ser gravado

**Retorna:**

Nulo

**Exceções:**

- XMSEException
- MessageNotWritableException

*WriteInt -Número inteiro de gravação*

**Interface:**

```
void WriteInt(Int32 value);
```

Escreva um número inteiro no fluxo de mensagem como 4 bytes, primeiro byte de alta ordem.

**Parâmetros:**

**valor (entrada)**

O número inteiro a ser gravado.

**Retorna:**

Nulo

**Exceções:**

- XMSEException
- MessageNotWritableException

*WriteLong -Número inteiro longo de gravação*

**Interface:**

```
void WriteLong(Int64 value);
```

Grave um número inteiro longo no fluxo de mensagens como 8 bytes, primeiro byte de alta ordem.

**Parâmetros:**

**valor (entrada)**

O número inteiro longo a ser gravado.

**Retorna:**

Nulo

**Exceções:**

- XMSEException
- MessageNotWritableException

*WriteObject -Gravar Objeto*

**Interface:**

```
void WriteObject(Object value);
```

Grave um valor, com um tipo de dado especificado, no fluxo de mensagens.

**Parâmetros:****objectType (entrada)**

O valor, que deve ser um dos seguintes tipos de objeto:

Boolean  
Byte  
Byte[]  
Char  
Double  
Single  
Int32  
Int64  
Int16  
String

**valor (entrada)**

Uma matriz de bytes contendo o valor a ser gravado.

**comprimento (entrada)**

O número de bytes na matriz

**Retorna:**

Nulo

**Exceções:**

- XMSEException

*WriteShort -Escrever Número Inteiro Curto*

**Interface:**

```
void WriteShort(Int16 value);
```

Grave um número inteiro curto no fluxo de mensagens como 2 bytes, primeiro byte de alta ordem.

**Parâmetros:****valor (entrada)**

O número inteiro curto a ser gravado.

**Retorna:**

Nulo

**Exceções:**

- XMSEException
- MessageNotWritableException

*WriteString -Sequência de Gravação*

**Interface:**

```
void WriteString(String value);
```

Grave uma sequência no fluxo de mensagens.

**Parâmetros:****valor (entrada)**

Um objeto String encapsulando a sequência a ser gravada.

**Retorna:**

Nulo

### Exceções:

- XMSException
- MessageNotWritableException

### Propriedades e métodos herdados

As propriedades a seguir são herdadas da interface [IMessage](#):

[JMSCorrelationID](#), [JMSDeliveryMode](#), [JMSDestination](#), [JMSExpiration](#), [JMSMessageID](#), [JMSPriority](#), [JMSRedelivered](#), [JMSReplyTo](#), [JMSTimestamp](#), [JMSType](#), [Propriedades](#)

Os métodos a seguir são herdados da interface [IMessage](#):

[clearBody](#), [clearProperties](#), [PropertyExists](#)

Os métodos a seguir são herdados da interface [IPropertyContext](#):

[GetBooleanProperty](#), [GetByteProperty](#), [GetBytesProperty](#), [GetCharProperty](#), [GetDoubleProperty](#), [GetFloatProperty](#), [GetIntProperty](#), [GetLongProperty](#), [GetObjectProperty](#), [GetShortProperty](#), [GetStringProperty](#), [SetBooleanProperty](#), [SetByteProperty](#), [SetBytesProperty](#), [SetCharProperty](#), [SetDoubleProperty](#), [SetFloatProperty](#), [SetIntProperty](#), [SetLongProperty](#), [SetObjectProperty](#), [SetShortProperty](#), [SetStringProperty](#)

## ITextMessage

Uma mensagem de texto é uma mensagem cujo corpo compreende uma sequência.

### Hierarquia de herança:

```
IBM.XMS.IPropertyContext
|
+---- IBM.XMS.IMessage
      |
      +---- IBM.XMS.ITextMessage
```

## .NET propriedades

*Texto-Obter e Configurar Texto*

### Interface:

```
String Text
{
    get;
    set;
}
```

Get e configure a sequência que forma o corpo da mensagem de texto.

Se necessário, XMS converte os caracteres na sequência na página de código local.

### Exceções:

- XMSException
- MessageNotReadableException
- MessageNotWritableException
- MessageEOFException

### Propriedades e métodos herdados

As propriedades a seguir são herdadas da interface [IMessage](#):

JMSCorrelationID, JMSDeliveryMode, JMSDestination, JMSExpiration, JMSMessageID, JMSPriority, JMSRedelivered, JMSReplyTo, JMSTimestamp, JMSType, Propriedades

Os métodos a seguir são herdados da interface IMessage:

clearBody, clearProperties, PropertyExists

Os métodos a seguir são herdados da interface IPropertyContext:

GetBooleanProperty, GetByteProperty, GetBytesProperty, GetCharProperty, GetDoubleProperty, GetFloatProperty, GetIntProperty, GetLongProperty, GetObjectProperty, GetShortProperty, GetStringProperty, SetBooleanProperty, SetByteProperty, SetBytesProperty, SetCharProperty, SetDoubleProperty, SetFloatProperty, SetIntProperty, SetLongProperty, SetObjectProperty, SetShortProperty, SetStringProperty

## TransactionInProgressException

XMS lançará essa exceção se um aplicativo solicitar uma operação que não seja válida porque uma transação está em andamento

### Hierarquia de herança:

```
IBM.XMS.XMSException
|
+----IBM.XMS.XMSException
|
+----IBM.XMS.TransactionInProgressException
```

### Propriedades e métodos herdados

Os métodos a seguir são herdados da interface XMSException:

GetErrorCódigo, GetLinkedExceção

## TransactionRolledBackException

XMS lança esta exceção se um aplicativo chamar `Session.commit()` para confirmar a transação atual, mas a transação será, então, revertida.

### Hierarquia de herança:

```
IBM.XMS.XMSException
|
+----IBM.XMS.XMSException
|
+----IBM.XMS.TransactionRolledBackException
```

### Propriedades e métodos herdados

Os métodos a seguir são herdados da interface XMSException:

GetErrorCódigo, GetLinkedExceção

## XMSException

Se XMS detectar um erro ao processar uma chamada para um método .NET, XMS lançará uma exceção. Uma exceção é um objeto que contém informações sobre o erro.

### Hierarquia de herança:

```
System.Exception
|
+----IBM.XMS.XMSException
```

Há diferentes tipos de exceção XMS , e um objeto XMSException é apenas um tipo de exceção. Entretanto, a classe XMSException é uma superclasse das outras classes de exceção XMS . XMS lança um objeto XMSException em situações em que nenhum dos outros tipos de exceção é apropriado.

## **.NET propriedades**

*ErrorCode* -Obter Código de Erro

### **Interface:**

```
public String ErrorCode
{
    get {return errorCode_;}
}
```

Obter o código de erro

### **Exceções:**

- XMSException

*LinkedException* -Obter Exceção Vinculada

### **Interface:**

```
public Exception LinkedException
{
    get { return linkedException_;}
    set { linkedException_ = value;}
}
```

Obter a próxima exceção na cadeia de exceção.

O método retornará um nulo se não houver mais exceções na cadeia.

### **Exceções:**

- XMSException

## **XMSFactoryFactory**

Se um aplicativo não estiver usando objetos administrados, utilize essa classe para criar connection factories, filas e tópicos..

### **Hierarquia de herança:**

Nenhum

## **.NET propriedades**

*Metadados*-Recuperar metadados

### **Interface:**

```
IConnectionMetaData MetaData
```

Obtenha os metadados apropriados para o tipo de conexão do objeto XMSFactoryFactory .

### **Exceções:**

Nenhum

### **Métodos**

## *CreateConnectionFactory-Criar Connection Factory*

### **Interface:**

```
ConnectionFactory CreateConnectionFactory();
```

Crie um objeto ConnectionFactory do tipo declarado.

### **Parâmetros:**

Nenhum

### **Retorna:**

O objeto ConnectionFactory .

### **Exceções:**

- XMSEException

## *CreateQueue -Criar Fila*

### **Interface:**

```
IDestination CreateQueue(String name);
```

Crie um objeto de Destino para representar uma fila no servidor de sistema de mensagens

Este método não cria a fila no servidor de mensagens. Deve-se criar a fila antes que um aplicativo possa chamar esse método

### **Parâmetros:**

#### **nome (entrada)**

Um objeto String encapsulando o nome da fila ou encapsulando um identificador uniforme de recursos (URI) que identifica a fila.

### **Retorna:**

O objeto de Destino que representa a fila

### **Exceções:**

- XMSEException

## *CreateTopic -Criar Tópico*

### **Interface:**

```
IDestination CreateTopic(String name);
```

Crie um objeto de Destino para representar um tópico

### **Parâmetros:**

#### **nome (entrada)**

Um objeto String encapsulando o nome do tópico ou encapsulando um identificador uniforme de recursos (URI) que identifica o tópico.

### **Retorna:**

O objeto de Destino que representa o tópico

### **Exceções:**

- XMSEException

*GetInstance -Obtenha uma instância de XMSFactoryFactory*

**Interface:**

```
static XMSFactoryFactory GetInstance(int connectionType);
```

Criar uma instância de XMSFactoryFactory. Um aplicativo XMS usa um objeto XMSFactoryFactory para obter uma referência a um objeto ConnectionFactory apropriado para o tipo de protocolo necessário. Esse objeto ConnectionFactory pode então produzir conexões somente para esse tipo de protocolo.

**Parâmetros:**

**connectionType (entrada)**

O tipo de conexão para o qual o objeto ConnectionFactory produz conexões:

- XMSC.CT\_WPM
- XMSC.CT\_RTT
- XMSC.CT\_WMQ

**Retorna:**

O objeto XMSFactoryFactory dedicado ao tipo de conexão declarado.

**Exceções:**

- Exceção NotSupportedException

## Propriedades de objetos XMS

Esta seção documenta as propriedades de objeto definidas por XMS

Esta seção contém informações sobre os seguintes tipos de objeto:

- [“Propriedades da Conexão” na página 2101](#)
- [“Propriedades de ConnectionFactory” na página 2101](#)
- [“Propriedades de Dados ConnectionMeta” na página 2106](#)
- [“Propriedades de Destino” na página 2106](#)
- [“Propriedades de InitialContext” na página 2108](#)
- [“Propriedades de Mensagem” na página 2109](#)
- [“Propriedades de MessageConsumer” na página 2114](#)
- [“Propriedades do MessageProducer” na página 2114](#)
- [“Propriedades da Sessão.” na página 2114](#)

A descrição de cada tipo de objeto lista as propriedades de um objeto do tipo especificado e fornece uma descrição simples de cada propriedade..

Esta seção também fornece uma definição de cada propriedade (consulte [“Definições de propriedades” na página 2114](#))

Se um aplicativo definir suas próprias propriedades dos objetos descritos nesta seção, ele não causará um erro, mas poderá causar resultados imprevisíveis

**Nota:** Os nomes e valores da propriedade nesta seção são mostrados no formulário XMSC . NAME, que é o formulário usado para C e C + +. No entanto, no .NET, o formulário do nome da propriedade pode ser XMSC . NAME ou XMSC \_NAME, dependendo de como você está usando:

- Se você estiver especificando uma propriedade, o nome da propriedade deverá estar no formato XMSC . NAME conforme mostrado no exemplo a seguir:

```
cf.SetStringProperty(XMSC.WMQ_CHANNEL, "DOTNET.SVRCONN");
```



- Se você estiver especificando uma cadeia, o nome da propriedade deverá estar no formato `XMSC_NAME`, conforme mostrado no exemplo a seguir:

```
cf.SetStringProperty("XMSC_WMQ_CHANNEL", "DOTNET.SVRCONN");
```

No .NET, os nomes e valores de propriedades são fornecidos como constantes na classe XMSC. Essas constantes identificam sequências e seriam usadas por qualquer aplicativo XMS .NET. Se você estiver usando essas constantes predefinidas, os nomes e valores de propriedades estarão no formato `XMSC.NAME`, portanto, por exemplo, você usaria `XMSC.USERID`, em vez de `XMSC_USERID`.

Os tipos de dados também estão no formato usado para C/C++. É possível localizar os valores correspondentes para .NET em [Tipos de Dados para .NET](#).

## Propriedades da Conexão

Uma visão geral das propriedades do objeto de Conexão, com links para informações de referência mais detalhadas

Nome da propriedade	Descrição
<a href="#">"XMSC_WMQ_RESOLVED_QUEUE_MANAGER" na página 2150</a>	Essa propriedade é usada para obter o nome do gerenciador de filas ao qual ele está conectado
<a href="#">"XMSC_WMQ_RESOLVED_QUEUE_MANAGER_ID" na página 2150</a>	Essa propriedade é preenchida com o ID do Gerenciador de Filas após a conexão.
<a href="#">XMSC_WPM_CONNECTION_PROTOCOL</a>	O protocolo de comunicações usado para a conexão com o mecanismo do sistema de mensagens. Esta propriedade é somente leitura.
<a href="#">XMSC_WPM_HOST_NAME</a>	O nome do host ou o endereço IP do sistema que contém o mecanismo do sistema de mensagens para o qual o aplicativo está conectado. Esta propriedade é somente leitura.
<a href="#">XMSC_WPM_ME_NAME</a>	O nome do mecanismo do sistema de mensagens para o qual o aplicativo está conectado. Esta propriedade é somente leitura.
<a href="#">XMSC_WPM_PORT</a>	O número da porta atendida pelo mecanismo do sistema de mensagens para o qual o aplicativo está conectado. Esta propriedade é somente leitura.

Um objeto Connection também possui propriedades somente leitura que são derivadas das propriedades do connection factory que foi usado para criar a conexão. Essas propriedades são derivadas não apenas das propriedades do connection factory que foram definidas no momento em que a conexão foi criada, mas também dos valores padrão das propriedades não configuradas. As propriedades incluem apenas aquelas relevantes para o tipo de servidor de sistema de mensagens ao qual o aplicativo está conectado. Os nomes das propriedades são iguais aos nomes das propriedades do connection factory.

## Propriedades de ConnectionFactory

Uma visão geral das propriedades do objeto ConnectionFactory, com links para informações de referência mais detalhadas.

Tabela 873. Propriedades de ConnectionFactory

Nome da propriedade	Descrição
<a href="#">“XMSC_ASYNC_EXCEPÇÕES”</a> na página 2125	Essa propriedade determina se o XMS informa um ExceptionListener apenas quando uma conexão é quebrada ou quando qualquer exceção ocorre de forma assíncrona para uma chamada da API do XMS. Essa propriedade se aplica a todas as Conexões criadas por meio dessa ConnectionFactory que possui um ExceptionListener registrado.
<b>V 9.3.0</b> <a href="#">“XMSC_WMQ_BALANCING_APPLICATION_TYPE”</a> na página 2133	Tipo de opção de balanceamento
<b>V 9.3.0</b> <a href="#">“XMSC_WMQ_BALANCING_OPTIONS”</a> na página 2134	Opções de balanceamento configuradas pelo aplicativo de emissão
<b>V 9.3.0</b> <a href="#">“XMSC_WMQ_BALANCING_TIMEOUT”</a> na página 2134	Tempo limite após o qual o rebalanceamento pode interromper a atividade do aplicativo
<a href="#">XMSC_CLIENT_ID</a>	O identificador do cliente para uma conexão.
<a href="#">XMSC_CONNECTION_TYPE</a>	O tipo de servidor de mensagens para o qual um aplicativo se conecta.
<a href="#">XMSC_PASSWORD</a>	Uma senha que pode ser usada para autenticar o aplicativo quando ele tenta se conectar a um servidor de mensagens.
<a href="#">“XMSC_RTT_BROKER_PING_INTERVAL”</a> na página 2130	O intervalo de tempo, em milissegundos, após o qual o XMS .NET verifica a conexão com um servidor do sistema de mensagens Real Time para detectar qualquer atividade.
<a href="#">XMSC_RTT_CONNECTION_PROTOCOL</a>	O protocolo de comunicações usado para uma conexão em tempo real com um broker.
<a href="#">XMSC_RTT_HOST_NAME</a>	O nome do host ou o endereço IP do sistema no qual um broker é executado.
<a href="#">XMSC_RTT_LOCAL_ADDRESS</a>	O nome do host ou o endereço IP da interface de rede local a ser usada para uma conexão em tempo real com um broker.
<a href="#">XMSC_RTT_MULTICAST</a>	A configuração multicast para um connection factory ou destino.
<a href="#">XMSC_RTT_PORT</a>	O número da porta na qual um broker atende às solicitações recebidas.
<a href="#">XMSC_USERID</a>	Um identificador de usuário que pode ser usado para autenticar o aplicativo quando ele tenta se conectar a um servidor de mensagens.
<a href="#">XMSC_WMQ_BROKER_CONTROLQ</a>	O nome da fila de controle usada pelo broker.
<a href="#">XMSC_WMQ_BROKER_PUBQ</a>	O nome da fila monitorada por um broker na qual os aplicativos enviam mensagens que publicam.
<a href="#">XMSC_WMQ_BROKER_QMGR</a>	O nome do gerenciador de filas ao qual um broker está conectado.

<i>Tabela 873. Propriedades de ConnectionFactory (continuação)</i>	
<b>Nome da propriedade</b>	<b>Descrição</b>
<u>XMSC_WMQ_BROKER_SUBQ</u>	O nome da fila de assinantes para um consumidor de mensagens não durável.
<u>XMSC_WMQ_BROKER_VERSION</u>	O tipo de corretor usado pelo aplicativo para uma conexão ou para o destino.
<u>“XMSC_WMQ_CCDTURL” na página 2136</u>	Um Localizador Uniforme de Recursos (URL) que identifica o nome e o local do arquivo que contém a tabela de definição de canal do cliente e também especifica como o arquivo pode ser acessado.
<u>XMSC_WMQ_CHANEXO L</u>	O nome do canal a ser usado para uma conexão.
<u>“XMSC_WMQ_CLIENT_RECONNECT_OPTIONS” na página 2137</u>	Esta propriedade especifica as opções de reconexão do cliente para novas conexões criadas por este factory..
<u>“XMSC_WMQ_CLIENT_RECONNECT_TIMEOUT” na página 2138</u>	Esta propriedade especifica a duração de tempo, em segundos, que uma conexão do cliente tenta reconectar.
<u>XMSC_WMQ_CONNECTION_MODE</u>	O modo pelo qual um aplicativo se conecta a um gerenciador de filas.
<u>“XMSC_WMQ_CONNECTION_NAME_LIST” na página 2138</u>	Esta propriedade especifica os hosts aos quais o cliente tenta se reconectar depois que sua conexão é interrompida
<u>XMSC_WMQ_FAIL_IF QUIESCE</u>	Se as chamadas para determinados métodos falharão se o gerenciador de filas ao qual o aplicativo está conectado estiver em um estado quiesce.
<u>XMSC_WMQ_HOST_NAME</u>	O nome do host ou o endereço IP do sistema no qual um gerenciador de filas é executado.
<u>XMSC_WMQ_LOCAL_ADDRESS</u>	Para uma conexão com um gerenciador de filas, essa propriedade especifica a interface de rede local a ser usada, a porta local ou o intervalo de portas locais a serem usadas ou ambos.
<u>XMSC_WMQ_MESSAGE_SELECTION</u>	Determina se a seleção de mensagens é feita por XMS cliente ou pelo broker.
<u>XMSC_WMQ_MSG_BATCH_SIZE</u>	O número máximo de mensagens a serem recuperadas de uma fila em um lote ao usar a entrega de mensagem assíncrona.
<u>XMSC_WMQ_POLLING_INTERVAL</u>	Se cada listener de mensagem dentro de uma sessão não tiver mensagens adequadas em sua fila, este valor será o intervalo máximo, em milissegundos, que decorrerá antes que cada listener da mensagem tente novamente obter uma mensagem de sua fila.
<u>“XMSC_WMQ_PROVIDER_VERSION” na página 2147</u>	A versão, liberação, nível de modificação e fix pack do gerenciador de filas ao qual o aplicativo pretende se conectar.
<u>XMSC_WMQ_PORT</u>	O número da porta na qual um gerenciador de filas atende às solicitações recebidas.
<u>XMSC_WMQ_PUB_ACK_INTERVAL</u>	O número de mensagens publicadas por um publicador antes do cliente XMS solicitar uma confirmação do broker.

<i>Tabela 873. Propriedades de ConnectionFactory (continuação)</i>	
<b>Nome da propriedade</b>	<b>Descrição</b>
<a href="#">“XMSC_WMQ_PUT_ASYNC_ALLOWED” na página 2143</a>	Essa propriedade determina se os produtores de mensagens têm permissão para usar as postagens assíncronas para enviar mensagens para esse destino.
<a href="#">XMSC_WMQ_QMGR_CCSD</a>	O identificador (CCSID) do conjunto de caractere codificado, ou página de códigos, no qual os campos de dados de caractere definidos na Message Queue Interface (MQI) são trocados entre o cliente XMS e o cliente IBM MQ.
<a href="#">XMSC_WMQ_QUEUE_MANAGER</a>	O nome do gerenciador de filas para conexão.
<a href="#">XMSC_WMQ_RECEIVE_EXIT</a>	Identifica uma saída de recebimento do canal para ser executada.
<a href="#">XMSC_WMQ_RECEIVE_EXIT_INIT</a>	Os dados do usuário que são transmitidos para uma saída de recebimento de canal quando ela é chamada.
<a href="#">XMSC_WMQ_SECURITY_EXIT</a>	Identifica uma saída de segurança do canal.
<a href="#">XMSC_WMQ_SECURITY_EXIT_INIT</a>	Os dados do usuário que são transmitidos para uma saída de segurança do canal quando ela é chamada.
<a href="#">“XMSC_WMQ_SEND_CHECK_COUNT” na página 2151</a>	O número de chamadas de envio a serem permitidas entre a verificação de erros de postagem assíncrona, dentro de uma única sessão XMS não transacionada.
<a href="#">XMSC_WMQ_SEND_EXIT</a>	Identifica uma saída de envio de canal.
<a href="#">XMSC_WMQ_SEND_EXIT_INIT</a>	Os dados do usuário que são transmitidos para as saídas de envio do canal quando são chamadas.
<a href="#">“XMSC_WMQ_SHARE_CONV_ALLOWED” na página 2151</a>	Se uma conexão do cliente pode compartilhar seu soquete com outras conexões do XMS de nível superior do mesmo processo para o mesmo gerenciador de filas, se as definições de canal corresponderem. Essa propriedade é fornecida para permitir o isolamento completo de Conexões em soquetes separados, se necessário para desenvolvimento de aplicativos, manutenção ou razões operacionais.
<a href="#">XMSC_WMQ_SSL_CERT_STORES</a>	Os locais dos servidores que retêm as listas de revogação de certificado (CRLs) a serem usadas em uma conexão SSL com um gerenciador de filas.
<a href="#">XMSC_WMQ_SSL_CIPHER_SPEC</a>	O nome da CipherSpec a ser usada em uma conexão segura com um gerenciador de filas.
<a href="#">XMSC_WMQ_SSL_CIPHER_SUITE</a>	O nome do CipherSuite a ser usado em uma conexão TLS para um gerenciador de filas. O protocolo usado para negociar a conexão segura depende do CipherSuite especificado.
<a href="#">XMSC_WMQ_SSL_CRYPT_HW</a>	Detalhes de configuração para o hardware criptográfico conectado ao sistema do cliente.

Tabela 873. Propriedades de ConnectionFactory (continuação)

Nome da propriedade	Descrição
<a href="#"><u>XMSC_WMQ_SSL_FIPS_REQUIRED</u></a>	O valor dessa propriedade determina se um aplicativo pode ou não usar conjuntos de cifras compatíveis não FIPS. Se essa propriedade for configurada como true, apenas algoritmos do FIPS serão usados para a conexão cliente-servidor.
<a href="#"><u>XMSC_WMQ_SSL_KEY_REPOSITORY</u></a>	O local do arquivo do banco de dados de chaves no qual chaves e certificados são armazenados.
<a href="#"><u>XMSC_WMQ_SSL_KEY_RESETCOUNT</u></a>	O KeyResetCount representa o número total de bytes não criptografados enviados e recebidos dentro de uma conversa SSL antes de a chave secreta ser renegociada.
<a href="#"><u>XMSC_WMQ_SSL_PEER_NAME</u></a>	O nome do peer a ser usado em uma conexão SSL com um gerenciador de filas.
<a href="#"><u>XMSC_WMQ_SYNCPOINT_ALL_GETS</u></a>	Se todas as mensagens devem ser recuperadas de filas dentro do controle de ponto de sincronização.
<a href="#"><u>"XMSC_WMQ_TARGET_CLIENT"</u></a> na página 2158	
<a href="#"><u>XMSC_WMQ_TEMP_Q_PREFIX</u></a>	O prefixo usado para formar o nome da fila dinâmica IBM MQ que é criada quando o aplicativo cria uma fila temporária XMS .
<a href="#"><u>XMSC_WMQ_TEMP_TOPIC_PREFIX</u></a>	Ao criar tópicos temporários, o XMS gera uma sequência de tópicos no formato "TEMP/TEMPTOPICPREFIX/unique_id" ou se essa propriedade contiver o valor padrão, essa sequência, "TEMP/unique_id", será gerada. Especificar um valor não vazio permite que as filas modelo específicas sejam definidas para criar as filas gerenciadas para assinantes de tópicos temporários criados sob essa conexão.
<a href="#"><u>XMSC_WMQ_TEMPORARY_MODEL</u></a>	O nome da fila modelo IBM MQ a partir da qual uma fila dinâmica é criada quando o aplicativo cria uma XMS fila temporária.
<a href="#"><u>XMSC_WPM_BUS_NAME</u></a>	Para um connection factory, o nome do barramento de integração de serviços ao qual o aplicativo se conecta ou, para um destino, o nome do barramento de integração de serviços no qual o destino existe.
<a href="#"><u>XMSC_WPM_CONNECTION_PROXIMIDADE</u></a>	A configuração de proximidade de conexão para a conexão.
<a href="#"><u>XMSC_WPM_DUR_SUB_HOME</u></a>	O nome do mecanismo do sistema de mensagens no qual todas as assinaturas duráveis para uma conexão ou um destino são gerenciadas.
<a href="#"><u>XMSC_WPM_LOCAL_ADDRESS</u></a>	Para uma conexão com um barramento de integração de serviços, essa propriedade especifica a interface de rede local a ser usada, a porta local ou o intervalo de portas locais a serem usados ou ambos.
<a href="#"><u>XMSC_WPM_NON_PERSISTENT_MAP</u></a>	O nível de confiabilidade de mensagens não persistentes que são enviadas usando a conexão.
<a href="#"><u>XMSC_WPM_PERSISTENT_MAP</u></a>	O nível de confiabilidade de mensagens persistentes que são enviadas usando a conexão.

*Tabela 873. Propriedades de ConnectionFactory (continuação)*

<b>Nome da propriedade</b>	<b>Descrição</b>
<a href="#"><u>XMSC_WPM_PROVIDER_ENDPOINTS</u></a>	Uma sequência de um ou mais endereços de terminal de servidores de autoinicialização.
<a href="#"><u>XMSC_WPM_TARGET_GROUP</u></a>	O nome de um grupo de destinos de mecanismos do sistema de mensagens.
<a href="#"><u>XMSC_WPM_TARGET_SIGNIFICATIVO</u></a>	O significado do grupo de destinos dos mecanismos do sistema de mensagens.
<a href="#"><u>XMSC_WPM_TARGET_TRANSPORT_CHAIN</u></a>	O nome da cadeia de transporte de entrada que o aplicativo deve usar para se conectar a um mecanismo do sistema de mensagens.
<a href="#"><u>XMSC_WPM_TARGET_TYPE</u></a>	O tipo do grupo de destinos de mecanismos do sistema de mensagens.
<a href="#"><u>XMSC_WPM_TEMP_Q_PREFIX</u></a>	O prefixo usado para formar o nome da fila temporária criada no barramento de integração de serviços quando o aplicativo cria uma XMS fila temporária.
<a href="#"><u>XMSC_WPM_TEMP_TOPIC_PREFIX</u></a>	O prefixo usado para formar o nome de um tópico temporário que é criado pelo aplicativo.

## Propriedades de Dados ConnectionMeta

Uma visão geral do objeto de dados ConnectionMeta, com links para informações de referência mais detalhadas.

*Tabela 874. Propriedades de Dados ConnectionMeta*

<b>Nome da propriedade</b>	<b>Descrição</b>
<a href="#"><u>XMSC_JMS_MAJOR_VERSION</u></a>	O número da versão principal da especificação JMS na qual XMS é baseado. Esta propriedade é somente leitura.
<a href="#"><u>XMSC_JMS_MINOR_VERSION</u></a>	O número da versão secundária da especificação JMS na qual XMS é baseado. Esta propriedade é somente leitura.
<a href="#"><u>XMSC_JMS_VERSION</u></a>	O identificador de versão da especificação JMS na qual XMS é baseado. Esta propriedade é somente leitura.
<a href="#"><u>XMSC_MAJOR_VERSION</u></a>	O número da versão do cliente do XMS Esta propriedade é somente leitura.
<a href="#"><u>XMSC_MINOR_VERSION</u></a>	O número da liberação do cliente do XMS Esta propriedade é somente leitura.
<a href="#"><u>XMSC_PROVIDER_NAME</u></a>	O provedor do cliente XMS . Esta propriedade é somente leitura.
<a href="#"><u>XMSC_VERSION</u></a>	O identificador de versão da cliXMSent. Essa propriedade é somente leitura.

## Propriedades de Destino

Uma visão geral das propriedades do objeto de Destino, com links para informações de referência mais detalhadas

<i>Tabela 875. Propriedades de Destino</i>	
<b>Nome da propriedade</b>	<b>Descrição</b>
<u>XMSC_DELIVERY_MODE</u>	O modo de entrega de mensagens enviadas para o destino.
<u>XMSC_PRIORITY</u>	A prioridade das mensagens enviadas para o destino.
<u>XMSC_RTT_MULTICAST</u>	A configuração multicast para um connection factory ou destino.
<u>XMSC_TIME_TO_LIVE</u>	O tempo de vida para mensagens enviadas para o destino.
<u>XMSC_WMQ_BROKER_VERSION</u>	O tipo de corretor usado pelo aplicativo para uma conexão ou para o destino.
<u>XMSC_WMQ_CCSD</u>	O identificador (CCSID) do conjunto de caracteres codificados, ou página de códigos, em que as cadeias de dados de caracteres no corpo de uma mensagem estão quando o cliente XMS encaminha a mensagem para o destino.
<u>XMSC_WMQ_DUR_SUBQ</u>	O nome da fila de assinantes para um assinante durável que está recebendo mensagens do destino.  <b>Nota:</b> Essa propriedade pode ser usada com a versão 2.0 do IBM Message Service Client for .NET , mas não tem efeito para um aplicativo conectado a um gerenciador de filas do IBM WebSphere MQ 7.0 , a menos que a propriedade XMSC_WWMQ_PROVIDER_VERSION do connection factory seja configurada para um número de versão menor que 7
<u>XMSC_WMQ_ENCODING</u>	Como os dados numéricos no corpo de uma mensagem são representados quando o cliente XMS encaminha a mensagem para o destino.
<u>XMSC_WMQ_FAIL_IF QUIESCE</u>	Se as chamadas para determinados métodos falharão se o gerenciador de filas ao qual o aplicativo está conectado estiver em um estado quiesce.
<u>“XMSC_WMQ_MESSAGE_BODY” na página 2141</u>	Essa propriedade determina se um aplicativo XMS processa o MQRFH2 de uma mensagem IBM MQ como parte da carga útil da mensagem (ou seja, como parte do corpo da mensagem).
<u>“XMSC_WMQ_MQMD_MESSAGE_CONTEXT” na página 2141</u>	Determina qual nível de contexto de mensagem deve ser configurado pelo aplicativo XMS . O aplicativo deve estar em execução com autoridade de contexto apropriado para esta propriedade entrar em vigor.
<u>“XMSC_WMQ_MQMD_READ_ENABLED” na página 2142</u>	Essa propriedade determina se um aplicativo XMS pode ou não extrair os valores dos campos MQMD
<u>“XMSC_WMQ_MQMD_WRITE_ENABLED” na página 2142</u>	Essa propriedade determina se um aplicativo XMS pode ou não configurar os valores dos campos MQMD
<u>“XMSC_WMQ_READ_AHEAD_ALLOWED” na página 2143</u>	Essa propriedade determina se os consumidores de mensagens e os navegadores de fila têm permissão para usar leitura antecipada para obter mensagens não persistentes, não transacionais desse destino em um buffer interno antes de recebê-las.



<i>Tabela 875. Propriedades de Destino (continuação)</i>	
<b>Nome da propriedade</b>	<b>Descrição</b>
<a href="#">“XMSC_WMQ_READ_AHEAD_CLOSE_POLICY” na página 2144</a>	Para mensagens que estão sendo entregues em um listener de mensagem assíncrona, essa propriedade determina o que acontece com as mensagens no buffer de leitura antecipada interno quando o consumidor de mensagens é fechado.
<a href="#">“XMSC_WMQ_RECEIVE_CCSD” na página 2149</a>	A propriedade de destino que configura o destino CCSID para a conversão de mensagens do gerenciador de filas. O valor é ignorado a menos que XMSC_WMQ_RECEIVE_CONVERSION seja configurado como WMQ_RECEIVE_CONVERSION_QMGR.
<a href="#">“XMSC_WMQ_RECEIVE_CONVERSION” na página 2149</a>	A propriedade de destino que determina se a conversão de dados será executada pelo gerenciador de filas.
<a href="#">XMSC_WMQ_TARGET_CLIENT</a>	Se as mensagens enviadas para o destino contêm um cabeçalho MQRFH2.
<a href="#">XMSC_WMQ_TEMP_TOPIC_PREFIX</a>	Ao criar tópicos temporários, o XMS gera uma sequência de tópicos no formato "TEMP/TEMPTOPICPREFIX/unique_id" ou se essa propriedade contiver o valor padrão, essa sequência, "TEMP/unique_id", será gerada. Especificar um valor não vazio permite que as filas modelo específicas sejam definidas para criar as filas gerenciadas para assinantes de tópicos temporários criados sob essa conexão.
<a href="#">XMSC_WPM_BUS_NAME</a>	Para um connection factory, o nome do barramento de integração de serviços ao qual o aplicativo se conecta ou, para um destino, o nome do barramento de integração de serviços no qual o destino existe.
<a href="#">XMSC_WPM_TOPIC_SPACE</a>	O nome do espaço de tópico que contém o tópico.

## Propriedades de InitialContext

Uma visão geral das propriedades do objeto InitialContext com links para informações de referência mais detalhadas.

<i>Tabela 876. Propriedades de InitialContext</i>	
<b>Nome da propriedade</b>	<b>Descrição</b>
<a href="#">XMSC_IC_PROVIDER_URL</a>	Usado para localizar o diretório de nomenclatura de JNDI para que o serviço de nomenclatura COS não precise estar no mesmo servidor que o serviço da web.
<a href="#">XMSC_IC_SECURITY_AUTHENTICATION</a>	Com base na Java Interface de contexto SECURITY_AUTHENTICATION Essa propriedade é aplicável apenas ao contexto de nomenclatura do COS
<a href="#">XMSC_IC_SECURITY_CREDENTIALS</a>	Com base na Java Interface de contexto SECURITY_CREDENTIALS Essa propriedade é aplicável apenas ao contexto de nomenclatura do COS
<a href="#">XMSC_IC_SECURITY_PRINCIPAL</a>	Com base na Java Interface de contexto SECURITY_PRINCIPAL Essa propriedade é aplicável apenas ao contexto de nomenclatura do COS



Tabela 876. Propriedades de InitialContext (continuação)

Nome da propriedade	Descrição
<a href="#">XMSC_IC_SECURITY_PROTOCOL</a>	Baseado na Java Interface de contexto SECURITY_PROTOCOL Essa propriedade é aplicável apenas ao contexto de nomenclatura do COS
<a href="#">XMSC_IC_URL</a>	Para os contextos LDAP e FileSystem, o endereço do repositório que contém objetos administrados. Para contextos de nomenclatura COS, o endereço do serviço da web que consulta os objetos no diretório.

## Propriedades de Mensagem

Uma visão geral das propriedades do objeto de Mensagem, com links para informações de referência mais detalhadas

Tabela 877. Propriedades de Mensagem

Nome da propriedade	Descrição
<a href="#">JMS_IBM_CHARACTER_SET</a>	O identificador (CCSID) do conjunto de caracteres codificados, ou página de códigos, em que as sequências de dados de caracteres no corpo da mensagem estão quando o cliente XMS encaminha a mensagem para seu destino desejado. Em XMS, esta propriedade possui um valor numérico e mapas para o CCSID. No entanto, esta propriedade é baseada em uma propriedade JMS portanto tem um valor de tipo string e mapas para o conjunto de caracteres Java que representa este CCSID numérico.
<a href="#">CODIFICAÇÃO DE JMS_IBM_ENCODING</a>	Como os dados numéricos no corpo da mensagem são representados quando o cliente XMS encaminha a mensagem para seu destino desejado.
<a href="#">JMS_IBM_EXCEPTIONMESSAGE</a>	Texto que descreve o motivo pelo qual a mensagem foi enviada para o destino de exceção. Esta propriedade é somente leitura.
<a href="#">JMS_IBM_ExceptionProblemDestination</a>	O nome do destino em que a mensagem estava antes de a mensagem ser enviada para o destino de exceção.
<a href="#">JMS_IBM_EXCEPTIONREASON</a>	Um código de razão que indica a razão pela qual a mensagem foi enviada para o destino de exceção.
<a href="#">JMS_IBM_EXCEPTIONTIMESTAMP</a>	O horário em que a mensagem foi enviada para o destino de exceção.
<a href="#">JMS_IBM_FEEDBACK</a>	Um código que indica a natureza de uma mensagem de relatório.
<a href="#">FORMATO JMS_IBM_FORMAT</a>	A natureza dos dados do aplicativo na mensagem.
<a href="#">JMS_IBM_LAST_MSG_IN_GROUP</a>	Indicar se a mensagem é a última mensagem em um grupo de mensagens.
<a href="#">JMS_IBM_MSGTYPE</a>	O tipo de mensagem.
<a href="#">JMS_IBM_PUTAPPLTYPE</a>	O tipo de aplicativo que enviou a mensagem.
<a href="#">JMS_IBM_PUTDATE</a>	A data em que a mensagem foi enviada.
<a href="#">JMS_IBM_PUTTIME</a>	O horário em que a mensagem foi enviada.

<i>Tabela 877. Propriedades de Mensagem (continuação)</i>	
<b>Nome da propriedade</b>	<b>Descrição</b>
<u>JMS_IBM_REPORT_COA</u>	Solicitar mensagens de relatório 'confirmar na chegada', especificando quantos dados do aplicativo da mensagem original devem ser incluídos em uma mensagem de relatório.
<u>JMS_IBM_REPORT_COD</u>	Solicitar mensagens de relatório 'confirmar na entrega', especificando quantos dados do aplicativo da mensagem original devem ser incluídos em uma mensagem de relatório.
<u>JMS_IBM_REPORT_DISCARD_MSG</u>	Solicitar que a mensagem seja descartada se não puder ser entregue a seu destino desejado.
<u>JMS_IBM_REPORT_EXCEPTION</u>	Solicitar mensagens de relatório de exceção, especificando quantos dados do aplicativo da mensagem original devem ser incluídos em uma mensagem de relatório.
<u>JMS_IBM_REPORT_EXPIRATION</u>	Solicitar mensagens de relatório de expiração, especificando quantos dados do aplicativo da mensagem original devem ser incluídos em uma mensagem de relatório.
<u>JMS_IBM_REPORT_NAN</u>	Solicitar mensagens de relatório de notificação de ação negativa.
<u>JMS_IBM_REPORT_PAN</u>	Solicitar mensagens de relatório de notificação de ação positiva.
<u>JMS_IBM_Report_Pass_Correl_ID</u>	Solicitar que o identificador de correlação de qualquer mensagem de relatório ou de resposta seja o mesmo que o identificador de correlação da mensagem original.
<u>JMS_IBM_REPORT_PASS_MSG_ID</u>	Solicitar que o identificador de mensagem de qualquer mensagem de relatório ou resposta seja o mesmo que o identificador de mensagem da mensagem original.
<u>JMS_IBM_RETAIN</u>	Configurar essa propriedade indica ao gerenciador de filas para tratar uma mensagem como Publicação retida.
<u>JMS_IBM_SYSTEM_MESSAGEID</u>	Um identificador que identifica a mensagem com exclusividade dentro do barramento de integração de serviços. Esta propriedade é somente leitura.
<u>JMSX_APPID</u>	O nome do aplicativo que enviou a mensagem.
<u>JMSX_DELIVERY_COUNT</u>	O número de tentativas de entregar a mensagem.
<u>JMSX_GROUPID</u>	O identificador do grupo de mensagens ao qual a mensagem pertence.
<u>JMSX_GROUPSEQ</u>	O número de sequência da mensagem dentro de um grupo de mensagens.
<u>JMSX_USERID</u>	O identificador de usuários associado ao aplicativo que enviou a mensagem.

### **Propriedades JMS\_IBM\_MQMD\***

IBM Message Service Client for .NET permite que aplicativos clientes leiam / gravem campos MQMD usando APIs. Ele também permite o acesso aos dados da mensagens do MQ Por padrão, o acesso ao

MQMD é desativado e deve ser ativado explicitamente pelo aplicativo usando as propriedades de Destino XMSC\_WMQ\_MQMD\_WRITE\_ENABLED e XMSC\_WMQ\_MQMD\_READ\_ENABLED. Essas duas propriedades são independentes entre si.

Todos os campos MQMD, exceto StructId e Version são expostos como propriedades adicionais do objeto de Mensagem e são prefixados JMS\_IBM\_MQMD.

As propriedades JMS\_IBM\_MQMD\* têm precedência mais alta sobre outras propriedades como JMS\_IBM\* descritas na tabela anterior.

## Enviando mensagens

Todos os campos MQMD, exceto StructId e Version, são representados. Essas propriedades referem-se apenas aos campos MQMD; quando uma propriedade ocorre tanto no MQMD quanto no cabeçalho MQRFH2, a versão no MQRFH2 não é configurada nem extraída. Qualquer uma dessas propriedades pode ser configurada, exceto JMS\_IBM\_MQMD\_BackoutCount. Qualquer valor configurado para JMS\_IBM\_MQMD\_BackoutCount é ignorado.

Se uma propriedade tiver um comprimento máximo e você fornecer um valor que é muito longo, o valor será truncado.

Para determinadas propriedades, deve-se também configurar a propriedade XMSC\_WMQ\_MQMD\_MESSAGE\_CONTEXT no objeto de Destino. O aplicativo deve estar em execução com autoridade de contexto apropriado para esta propriedade entrar em vigor. Se não configurar XMSC\_WMQ\_MQMD\_MESSAGE\_CONTEXT para um valor apropriado, o valor da propriedade será ignorado. Se você configurar XMSC\_WMQ\_MQMD\_MESSAGE\_CONTEXT para um valor apropriado, mas não tiver autoridade de contexto suficiente para o gerenciador de filas, uma exceção será emitida.. Propriedades que requerem valores específicos de XMSC\_WMQ\_MQMD\_MESSAGE\_CONTEXT são as seguintes.

As propriedades a seguir requerem que XMSC\_WMQ\_MQMD\_MESSAGE\_CONTEXT seja configurado como XMSC\_WMQ\_MDCTX\_SET\_IDENTITY\_CONTEXT ou XMSC\_WMQ\_MDCTX\_SET\_ALL\_CONTEXT:

- JMS\_IBM\_MQMD\_UserIdentifier
- JMS\_IBM\_MQMD\_AccountingToken
- JMS\_IBM\_MQMD\_ApplIdentityData

As seguintes propriedades requerem que XMSC\_WMQ\_MQMD\_MESSAGE\_CONTEXT seja configurado como XMSC\_WMQ\_MDCTX\_SET\_ALL\_CONTEXT:

- JMS\_IBM\_MQMD\_PutApplType
- JMS\_IBM\_MQMD\_PutApplName
- JMS\_IBM\_MQMD\_PutDate
- JMS\_IBM\_MQMD\_PutTime
- JMS\_IBM\_MQMD\_ApplOriginData

## Como receber mensagens

Todas essas propriedades estarão disponíveis em uma mensagem recebida se a propriedade XMSC\_WMQ\_MQMD\_READ\_ENABLED estiver configurada como true, independentemente das propriedades reais que o aplicativo de produção configurou como true. Um aplicativo não pode modificar as propriedades de uma mensagem recebida a menos que todas as propriedades sejam limpas primeiro, de acordo com a especificação JMS. A mensagem recebida pode ser transmitida sem modificar as propriedades.

**Nota:** Se seu aplicativo receber uma mensagem de um destino com a propriedade XMSC\_WMQ\_MQMD\_READ\_ENABLED configurada como true e encaminhá-la para um destino com XMSC\_WMQ\_MQMD\_WRITE\_ENABLED configurado como true, isso resultará em todos os valores do campo MQMD da mensagem recebida sendo copiados na mensagem encaminhada. Tabela de propriedades

<i>Tabela 878. Propriedades do objeto de Mensagem que representa os campos MQMD</i>		
<b>Propriedade</b>	<b>Descrição</b>	<b>tipo</b>
JMS_IBM_MQMD_Report	Opções para as mensagens de relatório	System.Int32
JMS_IBM_MQMD_MsgType	Tipo de mensagem	System.Int32
JMS_IBM_MQMD_Expiry	Tempo de vida da mensagem	System.Int32
JMS_IBM_MQMD_Feedback	Feedback ou código de razão	System.Int32
JMS_IBM_MQMD_Encoding	Codificação numérica de dados da mensagem	System.Int32
JMS_IBM_MQMD_CodedCharSetId	Identificador do conjunto de caracteres de dados da mensagem	System.Int32
JMS_IBM_MQMD_Format	Nome do formato dos dados da mensagem	System.String
JMS_IBM_MQMD_PRIORITY <b>Nota:</b> Se você designar um valor para JMS_IBM_MQMD_PRIORITY que não esteja no intervalo de 0 a 9, esse valor violará a especificação JMS.	Prioridade da mensagem	System.Int32
JMS_IBM_MQMD_Persistence	Persistência de mensagem	System.Int32
JMS_IBM_MQMD_MSGID <b>Nota:</b> A especificação JMS indica que o ID de mensagem deve ser configurado pelo provedor JMS e que deve ser exclusivo ou nulo. Se você designar um valor para JMS_IBM_MQMD_MsgId, esse valor será copiado para o JMSMessageID. Portanto, ele não é configurado pelo provedor JMS e pode não ser exclusivo: este valor viola a especificação JMS..	ID da Mensagem	matriz de byte <b>Nota:</b> O uso de propriedades de matriz de bytes em uma mensagem viola a especificação JMS
JMS_IBM_MQMD_CORRELID <b>Nota:</b> Se você designar um valor para JMS_IBM_MQMD_CORRELID que inicia com a sequência 'ID:', esse valor viola a especificação JMS.	Identificador de correlação	matriz de byte <b>Nota:</b> O uso de propriedades de matriz de bytes em uma mensagem viola a especificação JMS
JMS_IBM_MQMD_BackoutCount	contador de backout	System.Int32
JMS_IBM_MQMD_ReplyToQ	Nome da fila de resposta	System.String
JMS_IBM_MQMD_ReplyToQMgr	Nome do gerenciador de filas de resposta	System.String
JMS_IBM_MQMD_UserIdentifier	Identificador de usuários	System.String
JMS_IBM_MQMD_AccountingToken	Símbolo de contabilidade	matriz de byte <b>Nota:</b> O uso de propriedades de matriz de bytes em uma mensagem viola a especificação JMS

*Tabela 878. Propriedades do objeto de Mensagem que representa os campos MQMD (continuação)*

<b>Propriedade</b>	<b>Descrição</b>	<b>tipo</b>
JMS_IBM_MQMD_ApplIdentityData	dados do aplicativo relacionados à identidade	System.String
JMS_IBM_MQMD_PutApplType	Tipo de aplicativo que coloca a mensagem	System.Int32
JMS_IBM_MQMD_PutApplName	Nome do aplicativo que colocou a mensagem	System.String
JMS_IBM_MQMD_PutDate	Data quando a mensagem foi colocada	System.String
JMS_IBM_MQMD_PutTime	Hora quando a mensagem foi colocada	System.String
JMS_IBM_MQMD_ApplOriginData	Os dados do aplicativo relacionados à origem	System.String
JMS_IBM_MQMD_GroupId	Identificador de grupo	matriz de byte <b>Nota:</b> O uso de propriedades de matriz de bytes em uma mensagem viola a especificação JMS
JMS_IBM_MQMD_MsgSeqNumber	Número de sequência da mensagem local no grupo	System.Int32
JMS_IBM_MQMD_Offset	Deslocamento dos dados na mensagem física a partir do início da mensagem lógica	System.Int32
JMS_IBM_MQMD_MsgFlags	Sinalizadores de mensagem	System.Int32
JMS_IBM_MQMD_OriginalLength	Comprimento da mensagem original	System.Int32

Consulte [MQMD](#) para obter detalhes adicionais..

## Exemplos

Este exemplo resulta em uma mensagem sendo colocada em uma fila ou em um tópico com MQMD.UserIdentifier configurado como "JoeBloggs".

```
// Create a ConnectionFactory, connection, session, producer, message
// ...

// Create a destination
// ...

// Enable MQMD write
dest.setBooleanProperty(XMSC_WMQ_MQMD_WRITE_ENABLED,
    XMSC_WMQ_MQMD_WRITE_ENABLED_YES);

// Optionally, set a message context if applicable for this MD field
dest.setIntProperty(XMSC_WMQ_MQMD_MESSAGE_CONTEXT,
    XMSC_WMQ_MDCTX_SET_IDENTITY_CONTEXT);

// On the message, set property to provide custom UserId
msg.setStringProperty(JMS_IBM_MQMD_USERIDENTIFIER, "JoeBloggs");

// Send the message
// ...
```

É necessário configurar XMSC\_WMQ\_MQMD\_MESSAGE\_CONTEXT antes de configurar JMS\_IBM\_MQMD\_USERIDENTIFIER. Para obter mais informações sobre o uso de XMSC\_WMQ\_MQMD\_MESSAGE\_CONTEXT, consulte Propriedades do objeto Message.

Da mesma forma, é possível extrair o conteúdo dos campos do MQMD configurando XMSC\_WMQ\_MQMD\_READ\_ENABLED como true antes de receber uma mensagem e, em seguida, usando os métodos get da mensagem, como a propriedade getString. As propriedades recebidas são somente leitura.

Este exemplo resulta no campo de valor contendo o valor do MQMD MQMD.ApplIdentityData campo de uma mensagem obtido de uma fila ou de um tópico.

```
// Create a ConnectionFactory, connection, session, consumer
// ...

// Create a destination
// ...

// Enable MQMD read
dest.setBooleanProperty(XMSC_WMQ_MQMD_READ_ENABLED, XMSC_WMQ_MQMD_READ_ENABLED_YES);

// Receive a message
// ...

// Get required MQMD field value using a property
System.String value = rcvMsg.getStringProperty(JMS_IBM_MQMD_APPLIDENTITYDATA);
```

## Propriedades de MessageConsumer

Uma visão geral das propriedades do objeto MessageConsumer com links para informações de referência mais detalhadas.

<i>Tabela 879. Propriedades de MessageConsumer</i>	
Nome da propriedade	Descrição
<a href="#">XMSC_IS_SUBSCRIPTION_MULTICAST</a>	Indica se as mensagens estão sendo entregues para o consumidor de mensagens usando WebSphere MQ Multicast Transport. Esta propriedade é somente leitura.
<a href="#">XMSC_IS_SUBSCRIPTION_RELIABLE_MULTICAST</a>	Indica se as mensagens estão sendo entregues ao consumidor de mensagem usando WebSphere MQ Multicast Transport com uma qualidade de serviço confiável. Esta propriedade é somente leitura.

Consulte [.As propriedades NET de IMessageConsumer](#) para obter mais detalhes.

## Propriedades do MessageProducer

Uma visão geral das propriedades do objeto MessageProducer, com links para informações de referência mais detalhadas.

Consulte [.Propriedades NET de IMessageProducer](#) para obter mais detalhes.

## Propriedades da Sessão.

Uma visão geral das propriedades do objeto de Sessão, com links para informações de referência mais detalhadas.

Consulte [.Propriedades NET de ISession](#) para obter mais detalhes.

## Definições de propriedades

Esta seção fornece uma definição de cada propriedade de objeto.

Cada definição de propriedade inclui as seguintes informações:

- O tipo de dados da propriedade
- Os tipos de objeto que possuem a propriedade
- Para uma propriedade de Destino, o nome que pode ser usado em um URI (Identificador Uniforme de Recursos)
- Uma descrição mais detalhada da propriedade
- Os valores válidos da propriedade
- O valor padrão da propriedade

As propriedades cujos nomes começam com um dos seguintes prefixos são relevantes apenas para o tipo de conexão especificado:

#### **XMSC\_RTT**

As propriedades são relevantes apenas para uma conexão em tempo real com um broker Os nomes das propriedades são definidos como constantes nomeadas no arquivo de cabeçalho `xmsc_rtt.h`

#### **XMSC\_WMQ**

As propriedades são relevantes apenas quando um aplicativo se conecta a um gerenciador de fila do IBM MQ Os nomes das propriedades são definidos como constantes nomeadas no arquivo de cabeçalho `xmsc_wmq.h`

#### **XMSC\_WPM**

As propriedades são relevantes apenas quando um aplicativo se conecta a um barramento de integração de serviços do WebSphere Os nomes das propriedades são definidos como constantes nomeadas no arquivo de cabeçalho `xmsc_wpm.h`

Salvo indicação em contrário em suas definições, as propriedades restantes são relevantes para todos os tipos de conexão. Os nomes das propriedades são definidos como constantes nomeadas no arquivo de cabeçalho `xmsc.h` Propriedades cujos nomes começam com o prefixo JMSX são JMS propriedades definidas de uma mensagem e propriedades cujos nomes começam com o prefixo JMS\_IBM são IBM propriedades definidas de uma mensagem. Para obter mais informações sobre as propriedades de mensagens, consulte [Propriedades de uma XMS mensagem](#).

Salvo indicação em contrário em sua definição, cada propriedade é relevante nos domínios ponto a ponto e de assinatura de publicação.

Um aplicativo pode obter e configurar o valor de qualquer propriedade, a não ser que a propriedade seja designada como somente leitura.

### **JMS\_IBM\_CHARACTER\_SET**

#### **Tipo de dado:**

System.Int32

#### **Propriedade de:**

Mensagem

O identificador (CCSID) do conjunto de caracteres codificados, ou página de códigos, em que as sequências de dados de caracteres no corpo da mensagem estão quando o cliente XMS encaminha a mensagem para seu destino desejado. Em XMS, esta propriedade possui um valor numérico e mapas para o CCSID. No entanto, esta propriedade é baseada em uma propriedade JMS portanto tem um valor de tipo string e mapas para o conjunto de caracteres Java que representa este CCSID numérico. Essa propriedade substitui qualquer CCSID especificado para o destino pela propriedade [XMSC\\_WMQ\\_CCSD](#).

Por padrão, a propriedade não é configurada

Essa propriedade não é relevante quando um aplicativo se conecta a um barramento de integração de serviços

## ***CODIFICAÇÃO DE JMS\_IBM\_ENCODING***

**Tipo de dado:**

System.Int32

**Propriedade de:**

Mensagem

Como os dados numéricos no corpo da mensagem são representados quando o cliente XMS encaminha a mensagem para seu destino desejado. Essa propriedade substitui qualquer codificação especificada para o destino pela propriedade `XMSC_WMQ_ENCODING`.. A propriedade especifica a representação de números inteiros binários, números inteiros decimais empacotados e números de vírgula flutuante..

Os valores válidos da propriedade são os mesmos que os valores que podem ser especificados no campo **Encoding** de um descritor de mensagem

Um aplicativo pode usar as seguintes constantes nomeadas para configurar a propriedade:

<b>Constante nomeada</b>	<b>Significado</b>
<code>MQENC_INTEGER_NORMAL</code>	Codificação de número inteiro normal
<code>MQENC_INTEGER_REVERSED</code>	Codificação de número inteiro revertida
<code>MQENC_DECIMAL_NORMAL</code>	Codificação decimal compactada normal
<code>MQENC_DECIMAL_REVERSED</code>	Codificação decimal compactada revertida
<code>MQENC_FLOAT_IEEE_NORMAL</code>	Codificação de ponto flutuante IEEE normal
<code>MQENC_FLOAT_IEEE_REVERSED</code>	Codificação de ponto flutuante IEEE revertida
<code>MQENC_FLOAT_S390</code>	Codificação de ponto flutuante da arquitetura z/OS
<code>MQENC_NATIVE</code>	Codificação de máquina nativa

Para formar um valor para a propriedade, o aplicativo pode incluir três dessas constantes da seguinte forma:

- Uma constante cujo nome começa com `MQENC_INTEGER`, para especificar a representação de inteiros binários
- Uma constante cujo nome começa com `MQENC_DECIMAL`, para especificar a representação de números inteiros decimais compactados
- Uma constante cujo nome começa com `MQENC_FLOAT`, para especificar a representação de números de ponto flutuante

Como alternativa, o aplicativo pode configurar a propriedade para `MQENC_NATIVE`, cujo valor é dependente de ambiente.

Por padrão, a propriedade não é configurada

Essa propriedade não é relevante quando um aplicativo se conecta a um barramento de integração de serviços

## ***JMS\_IBM\_EXCEPTIONMESSAGE***

**Tipo de dado:**

Sequência

**Propriedade de:**

Mensagem

Texto que descreve o motivo pelo qual a mensagem foi enviada para o destino de exceção. Esta propriedade é somente leitura.

Essa propriedade é relevante apenas quando um aplicativo se conecta a um barramento de integração de serviços e recebe uma mensagem de um destino de exceções



## ***JMS\_IBM\_ExceptionProblemDestination***

**Tipo de dado:**

Sequência

**Propriedade de:**

Mensagem

O nome do destino em que a mensagem estava antes de a mensagem ser enviada para o destino de exceção.

Essa propriedade é relevante apenas quando um aplicativo se conecta a um barramento de integração de serviços e recebe uma mensagem de um destino de exceções

## ***JMS\_IBM\_EXCEPTIONREASON***

**Tipo de dado:**

System.Int32

**Propriedade de:**

Mensagem

Um código de razão que indica a razão pela qual a mensagem foi enviada para o destino de exceção.

Essa propriedade é relevante apenas quando um aplicativo se conecta a um barramento de integração de serviços e recebe uma mensagem de um destino de exceções

## ***JMS\_IBM\_EXCEPTIONTIMESTAMP***

**Tipo de dado:**

System.Int64

**Propriedade de:**

Mensagem

O horário em que a mensagem foi enviada para o destino de exceção.

O tempo é expresso em milissegundos desde 00:00:00 GMT de 1 de janeiro de 1970.

Essa propriedade é relevante apenas quando um aplicativo se conecta a um barramento de integração de serviços e recebe uma mensagem de um destino de exceções

## ***JMS\_IBM\_FEEDBACK***

**Tipo de dado:**

System.Int32

**Propriedade de:**

Mensagem

Um código que indica a natureza de uma mensagem de relatório.

Os valores válidos da propriedade são os códigos de feedback e de razão que podem ser especificados no campo **Feedback** de um descritor de mensagens.

Por padrão, a propriedade não é configurada

## ***FORMATO JMS\_IBM\_FORMAT***

**Tipo de dado:**

Sequência

**Propriedade de:**

Mensagem

A natureza dos dados do aplicativo na mensagem.

Os valores válidos da propriedade são os mesmos que os valores que podem ser especificados no campo **Format** de um descritor de mensagem

Por padrão, a propriedade não é configurada

Essa propriedade não é relevante quando um aplicativo se conecta a um barramento de integração de serviços

### ***JMS\_IBM\_LAST\_MSG\_IN\_GROUP***

**Tipo de dado:**

System.Boolean

**Propriedade de:**

Mensagem

Indicar se a mensagem é a última mensagem em um grupo de mensagens.

Configure a propriedade como true se a mensagem for a última mensagem em um grupo de mensagens.. Caso contrário, configure a propriedade como false, ou não configure a propriedades. Por padrão, a propriedade não é configurada

O valor true corresponde ao sinalizador de status MQMF\_LAST\_MSG\_IN\_GROUP, que pode ser especificado no campo **MsgFlags** de um descritor de mensagens.

Essa propriedade é ignorada no domínio de publicação / assinatura e não é relevante quando um aplicativo se conecta a um barramento de integração de serviços

### ***JMS\_IBM\_MSGTYPE***

**Tipo de dado:**

System.Int32

**Propriedade de:**

Mensagem

O tipo de mensagem.

Os valores válidos da propriedade são os seguintes:

<b>Valor válido</b>	<b>Significado</b>
MQMT_DATAGRAM	A mensagem é uma que não requer resposta.
MQMT_REQUEST	A mensagem é aquela que requer uma resposta.
MQMT_REPLY	A mensagem é uma mensagem de resposta
MQMT_REPORT	A mensagem é uma mensagem de relatório

Esses valores correspondem aos tipos de mensagens que podem ser especificados no campo **MsgType** de um descritor de mensagens

Por padrão, a propriedade não é configurada

Essa propriedade não é relevante quando um aplicativo se conecta a um barramento de integração de serviços

### ***JMS\_IBM\_PUTAPPLTYPE***

**Tipo de dado:**

System.Int32

**Propriedade de:**

Mensagem

O tipo de aplicativo que enviou a mensagem.

Os valores válidos da propriedade são os tipos de aplicativos que podem ser especificados no campo **PutApp1Type** de um descritor de mensagens

Por padrão, a propriedade não é configurada

Essa propriedade não é relevante quando um aplicativo se conecta a um barramento de integração de serviços

### ***JMS\_IBM\_PUTDATE***

**Tipo de dado:**

Sequência

**Propriedade de:**

Mensagem

A data em que a mensagem foi enviada.

Os valores válidos da propriedade são os mesmos que os valores que podem ser especificados no campo **PutDate** de um descritor de mensagem

Por padrão, a propriedade não é configurada

Essa propriedade não é relevante quando um aplicativo se conecta a um barramento de integração de serviços

### ***JMS\_IBM\_PUTTIME***

**Tipo de dado:**

Sequência

**Propriedade de:**

Mensagem

O horário em que a mensagem foi enviada.

Os valores válidos da propriedade são os mesmos que os valores que podem ser especificados no campo **PutTime** de um descritor de mensagem

Por padrão, a propriedade não é configurada

Essa propriedade não é relevante quando um aplicativo se conecta a um barramento de integração de serviços

### ***JMS\_IBM\_REPORT\_COA***

**Tipo de dado:**

System.Int32

**Propriedade de:**

Mensagem

Solicitar mensagens de relatório 'confirmar na chegada', especificando quantos dados do aplicativo da mensagem original devem ser incluídos em uma mensagem de relatório.

Os valores válidos da propriedade são os seguintes:

<b>Valor válido</b>	<b>Significado</b>
MQRO_COA	Solicite mensagens de relatório 'confirmar na chegada', sem dados do aplicativo da mensagem original incluídos em uma mensagem de relatório
MQRO_COA_WITH_DATA	Solicite 'confirmar na chegada' mensagens de relatório, com os primeiros 100 bytes de dados do aplicativo da mensagem original incluídos em uma mensagem de relatório
MQRO_COA_WITH_FULL_DATA	Solicite mensagens de relatório 'confirmar na chegada', com todos os dados do aplicativo da mensagem original incluídos em uma mensagem de relatório

Esses valores correspondem às opções do relatório que podem ser especificadas no campo **Report** de um descritor de mensagens Para obter mais informações sobre essas opções, consulte [Report \(MQLONG\)](#).

Por padrão, a propriedade não é configurada

### ***JMS\_IBM\_REPORT\_COD***

**Tipo de dado:**

System.Int32

**Propriedade de:**

Mensagem

Solicitar mensagens de relatório 'confirmar na entrega', especificando quantos dados do aplicativo da mensagem original devem ser incluídos em uma mensagem de relatório.

Os valores válidos da propriedade são os seguintes:

<b>Valor válido</b>	<b>Significado</b>
MQRO_COD	Solicite 'confirmar na entrega' mensagens de relatório, sem dados do aplicativo da mensagem original incluída em uma mensagem de relatório
MQRO_COD_WITH_DATA	Solicite 'confirmar na entrega' mensagens de relatório, com os primeiros 100 bytes de dados do aplicativo da mensagem original incluídos em uma mensagem de relatório
MQRO_COD_WITH_FULL_DATA	Solicite 'confirmar na entrega' mensagens de relatório, com todos os dados do aplicativo da mensagem original incluídos em uma mensagem de relatório

Esses valores correspondem às opções de relatório que podem ser especificadas no campo **Report** de um descritor de mensagens

Por padrão, a propriedade não é configurada

### ***JMS\_IBM\_REPORT\_DISCARD\_MSG***

**Tipo de dado:**

System.Int32

**Propriedade de:**

Mensagem

Solicitar que a mensagem seja descartada se não puder ser entregue a seu destino desejado.

Configure a propriedade para MQRO\_DISCARD\_MSG para solicitar que a mensagem seja descartada se não puder ser entregue para seu destino desejado. Se você requerer que a mensagem seja colocada em uma fila de devoluções ou enviada para um destino de exceções, não configure a propriedade Por padrão, a propriedade não é configurada

O valor MQRO\_DISCARD\_MSG corresponde a uma opção de relatório que pode ser especificada no campo **Report** de um descritor de mensagens

### ***JMS\_IBM\_REPORT\_EXCEPTION***

**Tipo de dado:**

System.Int32

**Propriedade de:**

Mensagem

Solicitar mensagens de relatório de exceção, especificando quantos dados do aplicativo da mensagem original devem ser incluídos em uma mensagem de relatório.

Os valores válidos da propriedade são os seguintes:

<b>Valor válido</b>	<b>Significado</b>
MQRO_EXCEPTION	Mensagens de relatório de exceção de solicitação, sem dados do aplicativo da mensagem original incluídos em uma mensagem de relatório
MQRO_EXCEPTION_WITH_DATA	Mensagens de relatório de exceção de solicitação, com os primeiros 100 bytes de dados do aplicativo da mensagem original incluídos em uma mensagem de relatório.
MQRO_EXCEPTION_WITH_FULL_DATA	Mensagens de relatório de exceção de solicitação, com todos os dados do aplicativo da mensagem original incluídos em uma mensagem de relatório

Esses valores correspondem às opções de relatório que podem ser especificadas no campo **Report** de um descritor de mensagens

Por padrão, a propriedade não é configurada

### ***JMS\_IBM\_REPORT\_EXPIRATION***

**Tipo de dado:**

System.Int32

**Propriedade de:**

Mensagem

Solicitar mensagens de relatório de expiração, especificando quantos dados do aplicativo da mensagem original devem ser incluídos em uma mensagem de relatório.

Os valores válidos da propriedade são os seguintes:

<b>Valor válido</b>	<b>Significado</b>
MQRO_EXPIRATION	Mensagens de relatório de expiração de solicitação, sem dados do aplicativo da mensagem original incluídos em uma mensagem de relatório
MQRO_EXPIRATION_WITH_DATA	Mensagens de relatório de expiração de solicitação, com os primeiros 100 bytes de dados do aplicativo da mensagem original incluídos em uma mensagem de relatório
MQRO_EXPIRATION_WITH_FULL_DATA	Mensagens de relatório de expiração de solicitação, com todos os dados do aplicativo da mensagem original incluídos em uma mensagem de relatório

Esses valores correspondem às opções de relatório que podem ser especificadas no campo **Report** de um descritor de mensagens

Por padrão, a propriedade não é configurada

### ***JMS\_IBM\_REPORT\_NAN***

**Tipo de dado:**

System.Int32

**Propriedade de:**

Mensagem

Solicitar mensagens de relatório de notificação de ação negativa.

Configure a propriedade para MQRO\_NAN para solicitar mensagens de notificação de ação negativa. Se você não precisar de mensagens de notificação de ação negativa, não configure a propriedade. Por padrão, a propriedade não é configurada

O valor MQRO\_NAN corresponde a uma opção de relatório que pode ser especificada no campo **Report** de um descritor de mensagens.

**JMS\_IBM\_REPORT\_PAN****Tipo de dado:**

System.Int32

**Propriedade de:**

Mensagem

Solicitar mensagens de relatório de notificação de ação positiva.

Configure a propriedade como MQRO\_PAN para solicitar mensagens de relatório de notificação de ação positiva. Se você não precisar de mensagens de notificação de ação positiva, não configure a propriedade. Por padrão, a propriedade não é configurada

O valor MQRO\_PAN corresponde a uma opção de relatório que pode ser especificada no campo **Report** de um descritor de mensagens.

**JMS\_IBM\_Report\_Pass\_Correl\_ID****Tipo de dado:**

System.Int32

**Propriedade de:**

Mensagem

Solicitar que o identificador de correlação de qualquer mensagem de relatório ou de resposta seja o mesmo que o identificador de correlação da mensagem original.

Os valores válidos da propriedade são os seguintes:

**Valor válido**

MQRO\_PASS\_CORREL\_ID

**Significado**

Solicitar que o identificador de correlação de qualquer mensagem de relatório ou de resposta seja o mesmo que o identificador de correlação da mensagem original.

MQRO\_COPY\_MSG\_ID\_TO\_CORREL\_ID

Solicite que o identificador de correlação de qualquer mensagem de relatório ou de resposta seja igual ao identificador de mensagem da mensagem original.

Esses valores correspondem às opções do relatório, que podem ser especificadas no campo **Report** de um descritor de mensagem

O valor padrão da propriedade é MQRO\_COPY\_MSG\_ID\_TO\_CORREL\_ID.

**JMS\_IBM\_REPORT\_PASS\_MSG\_ID****Tipo de dado:**

System.Int32

**Propriedade de:**

Mensagem

Solicitar que o identificador de mensagem de qualquer mensagem de relatório ou resposta seja o mesmo que o identificador de mensagem da mensagem original.

Os valores válidos da propriedade são os seguintes:

<b>Valor válido</b>	<b>Significado</b>
MQRO_PASS_MSG_ID	Solicitar que o identificador de mensagem de qualquer mensagem de relatório ou resposta seja o mesmo que o identificador de mensagem da mensagem original.
MQRO_NEW_MSG_ID	Solicite que um novo identificador de mensagem seja gerado para cada mensagem de relatório ou resposta.

Esses valores correspondem às opções do relatório que podem ser especificadas no campo [Relatório](#) de um descritor de mensagens

O valor padrão da propriedade é MQRO\_NEW\_MSG\_ID.

### ***JMS\_IBM\_RETAIN***

**Tipo de dado:**

System.Int32

**Propriedade de:**

Mensagem

Configurar essa propriedade indica ao gerenciador de filas para tratar uma mensagem como Publicação retida. Quando um assinante recebe mensagens de tópicos, ele pode receber mensagens adicionais imediatamente após a assinatura, além das mensagens recebidas em liberações anteriores. Essas mensagens são as publicações opcionais retidas para os tópicos inscritos. Para cada tópico correspondente à assinatura, se houver uma publicação retida, a publicação será disponibilizada para entrega ao consumidor de mensagens de assinatura.

RETAIN\_PUBLICATION é o único valor válido para esta propriedade. Por padrão, esta propriedade não é definida.

**Nota:** Esta propriedade é relevante apenas no domínio de publicação / assinatura

### ***JMS\_IBM\_SYSTEM\_MESSAGEID***

**Tipo de dado:**

Sequência

**Propriedade de:**

Mensagem

Um identificador que identifica a mensagem com exclusividade dentro do barramento de integração de serviços. Esta propriedade é somente leitura.

Esta propriedade é relevante apenas quando um aplicativo se conecta a um barramento de integração de serviços

### ***JMSX\_APPID***

**Tipo de dado:**

Sequência

**Propriedade de:**

Mensagem

O nome do aplicativo que enviou a mensagem.

Essa propriedade é a propriedade definida JMS com o JMS name JMSXAppID. Para obter mais informações sobre a propriedade, consulte o *Java Message Service Specification, Versão 1.1*.

Por padrão, a propriedade não é configurada

Essa propriedade não é válida para uma conexão em tempo real com um broker

### ***JMSX\_DELIVERY\_COUNT***

**Tipo de dado:**

System.Int32

**Propriedade de:**

Mensagem

O número de tentativas de entregar a mensagem.

Essa propriedade é a propriedade definida pelo JMS com o JMS name JMSXDeliveryCount Para obter mais informações sobre a propriedade, consulte *Java Message Service Especificação, Versão 1.1*.

Por padrão, a propriedade não é configurada

Essa propriedade não é válida para uma conexão em tempo real com um broker

### ***JMSX\_GROUPID***

**Tipo de dado:**

Sequência

**Propriedade de:**

Mensagem

O identificador do grupo de mensagens ao qual a mensagem pertence.

Essa propriedade é a propriedade definida JMS com o JMS nome JMSXGroupID. Para obter mais informações sobre a propriedade, consulte *Java Message Service Especificação, Versão 1.1*.

Por padrão, a propriedade não é configurada

Essa propriedade não é válida para uma conexão em tempo real com um broker

### ***JMSX\_GROUPSEQ***

**Tipo de dado:**

System.Int32

**Propriedade de:**

Mensagem

O número de sequência da mensagem dentro de um grupo de mensagens.

Essa propriedade é a propriedade definida JMS com o JMS nome JMSXGroupSeq. Para obter mais informações sobre a propriedade, consulte *Java Message Service Especificação, Versão 1.1*.

Por padrão, a propriedade não é configurada

Essa propriedade não é válida para uma conexão em tempo real com um broker

### ***JMSX\_USERID***

**Tipo de dado:**

Sequência

**Propriedade de:**

Mensagem

O identificador de usuários associado ao aplicativo que enviou a mensagem.

Essa propriedade é a propriedade definida JMS com o JMS name JMSXUserID. Para obter mais informações sobre a propriedade, consulte *Java Message Service Especificação, Versão 1.1*.

Por padrão, a propriedade não é configurada

Essa propriedade não é válida para uma conexão em tempo real com um broker



## ***XMSC\_ASYNC\_EXCEPÇÕES***

**Tipo de dado:**

System.Int32

**Propriedade de:**

ConnectionFactory

**Objetos aplicáveis:**

Nome longo da ferramenta de administração JMS: ASYNCEXCEPTION

Nome abreviado da ferramenta de administração JMS: AEX

Essa propriedade determina se o XMS informa um ExceptionListener apenas quando uma conexão é quebrada ou quando qualquer exceção ocorre de forma assíncrona para uma chamada da API do XMS. Essa propriedade se aplica a todas as Conexões criadas por meio dessa ConnectionFactory que possui um ExceptionListener registrado.

Os valores válidos para essa propriedade são:

***XMSC\_ASYNC\_EXCEPTIONS\_ALL***

Qualquer exceção detectada de forma assíncrona, fora do escopo de uma chamada de API síncrona, e todas as exceções de conexão interrompida são enviadas para o ExceptionListener

***XMSC\_ASYNC\_EXCEPTIONS\_CONNECTIONBROKEN***

Apenas exceções indicando uma conexão interrompida são enviadas para o ExceptionListener.

Quaisquer outras exceções que ocorram durante o processamento assíncrono não são relatadas para o ExceptionListenere, portanto, o aplicativo não é informado sobre essas exceções

Por padrão, essa propriedade é configurada para XMSC\_ASYNC\_EXCEPTIONS\_ALL

## ***XMSC\_CLIENT\_ID***

**Tipo de dado:**

Sequência

**Propriedade de:**

ConnectionFactory

**Objetos aplicáveis:**

Nome longo da ferramenta de administração JMS: CLIENTID

Nome abreviado da ferramenta de administração JMS: CID

O identificador do cliente para uma conexão.

Um identificador de cliente é usado apenas para suportar assinaturas duráveis no domínio de publicação / assinatura e é ignorado no domínio ponto a ponto. Para obter mais informações sobre como configurar identificadores de cliente, consulte [ConnectionFactoryes e objetos de Conexão](#)

Essa propriedade não é relevante para uma conexão em tempo real com um broker

## ***XMSC\_CONNECTION\_TYPE***

**Tipo de dado:**

System.Int32

**Propriedade de:**

ConnectionFactory

O tipo de servidor de mensagens para o qual um aplicativo se conecta.

Os valores válidos da propriedade são os seguintes:

<b>Valor válido</b>	<b>Significado</b>
XMSC_CT_RTT	Uma conexão em tempo real com um corretor.
XMSC_CT_WMQ	Uma conexão com um gerenciador de filas do IBM MQ

**Valor válido**

XMSC\_CT\_WPM

**Significado**

Uma conexão com um WebSphere Application Server service integration bus

Por padrão, a propriedade não é configurada

***XMSC\_DELIVERY\_MODE*****Tipo de dado:**

System.Int32

**Propriedade de:**

Destino

**Nome usado em um URI:**

persistência (para um destino IBM MQ)

deliveryMode (para um WebSphere destino do provedor de sistemas de mensagens padrão)

**Objetos aplicáveis:**

Nome longo da ferramenta de administração JMS: PERSISTENCE

Nome abreviado da ferramenta de administração JMS: PER

O modo de entrega de mensagens enviadas para o destino.

Os valores válidos da propriedade são os seguintes:

**Valor válido**

XMSC\_DELIVERY\_NOT\_PERSISTENT

**Significado**

Uma mensagem enviada para o destino é não persistente O modo de entrega padrão do produtor de mensagem, ou qualquer modo de entrega especificado na chamada Enviar, é ignorado. Se o destino for uma fila IBM MQ , o valor do atributo da fila *DefPersistence* também será ignorado.

XMSC\_DELIVERY\_PERSISTENT

Uma mensagem enviada para o destino é persistente O modo de entrega padrão do produtor de mensagem, ou qualquer modo de entrega especificado na chamada Enviar, é ignorado. Se o destino for uma fila IBM MQ , o valor do atributo da fila *DefPersistence* também será ignorado.

XMSC\_DELIVERY\_AS\_APP

Uma mensagem enviada para o destino tem o modo de entrega especificado na chamada Enviar. Se a chamada Enviar especificar nenhum modo de entrega, o modo de entrega padrão do produtor da mensagem será usado no lugar. Se o destino for uma fila IBM MQ , o valor do atributo da fila *DefPersistence* será ignorado.

XMSC\_DELIVERY\_AS\_DEST

Se o destino for uma fila IBM MQ , uma mensagem colocada na fila terá o modo de entrega especificado pelo valor do atributo da fila *DefPersistence*. O modo de entrega padrão do produtor de mensagem, ou qualquer modo de entrega especificado na chamada Enviar, é ignorado.

Se o destino não for uma fila IBM MQ , o significado será o mesmo que XMSC\_DELIVERY\_AS\_APP.

O valor-padrão é XMSC\_DELIVERY\_AS\_APP.

## ***XMSC\_IC\_PROVIDER\_URL***

**Tipo de dado:**

Sequência

**Propriedade de:**

InitialContext

Usado para localizar o diretório de nomenclatura de JNDI para que o serviço de nomenclatura COS não precise estar no mesmo servidor que o serviço da web.

## ***XMSC\_IC\_SECURITY\_AUTHENTICATION***

**Tipo de dado:**

Sequência

**Propriedade de:**

InitialContext

Com base na Java Interface de contexto SECURITY\_AUTHENTICATION Essa propriedade é aplicável apenas ao contexto de nomenclatura do COS

## ***XMSC\_IC\_SECURITY\_CREDENTIALS***

**Tipo de dado:**

Sequência

**Propriedade de:**

InitialContext

Com base na Java Interface de contexto SECURITY\_CREDENTIALS Essa propriedade é aplicável apenas ao contexto de nomenclatura do COS

## ***XMSC\_IC\_SECURITY\_PRINCIPAL***

**Tipo de dado:**

Sequência

**Propriedade de:**

InitialContext

Com base na Java Interface de contexto SECURITY\_PRINCIPAL Essa propriedade é aplicável apenas ao contexto de nomenclatura do COS

## ***XMSC\_IC\_SECURITY\_PROTOCOL***

**Tipo de dado:**

Sequência

**Propriedade de:**

InitialContext

Baseado na Java Interface de contexto SECURITY\_PROTOCOL Essa propriedade é aplicável apenas ao contexto de nomenclatura do COS

## ***XMSC\_IC\_URL***

**Tipo de dado:**

Sequência

**Propriedade de:**

InitialContext

Para os contextos LDAP e FileSystem, o endereço do repositório que contém objetos administrados.

Para os contextos LDAP e FileSystem, o endereço do repositório que contém objetos administrados.

## ***XMSC\_IS\_SUBSCRIPTION\_MULTICAST***

**Tipo de dado:**

System.Boolean

**Propriedade de:**

MessageConsumer

Indica se as mensagens estão sendo entregues para o consumidor de mensagens usando WebSphere MQ Multicast Transport. Esta propriedade é somente leitura.

O valor da propriedade será true se as mensagens estiverem sendo entregues ao consumidor de mensagens usando WebSphere MQ Multicast Transport. Caso contrário, o valor será false.

Essa propriedade é relevante apenas para uma conexão em tempo real com um broker

## ***XMSC\_IS\_SUBSCRIPTION\_RELIABLE\_MULTICAST***

**Tipo de dado:**

System.Boolean

**Propriedade de:**

MessageConsumer

Indica se as mensagens estão sendo entregues ao consumidor de mensagem usando WebSphere MQ Multicast Transport com uma qualidade de serviço confiável. Esta propriedade é somente leitura.

O valor da propriedade será true se as mensagens estiverem sendo entregues para o consumidor de mensagens usando WebSphere MQ Multicast Transport com uma qualidade de serviço confiável. Caso contrário, o valor será false.

Essa propriedade é relevante apenas para uma conexão em tempo real com um broker

## ***XMSC\_JMS\_MAJOR\_VERSION***

**Tipo de dado:**

System.Int32

**Propriedade de:**

ConnectionMetaData

O número da versão principal da especificação JMS na qual XMS é baseado. Esta propriedade é somente leitura.

## ***XMSC\_JMS\_MINOR\_VERSION***

**Tipo de dado:**

System.Int32

**Propriedade de:**

ConnectionMetaData

O número da versão secundária da especificação JMS na qual XMS é baseado. Esta propriedade é somente leitura.

## ***XMSC\_JMS\_VERSION***

**Tipo de dado:**

Sequência

**Propriedade de:**

ConnectionMetaData

O identificador de versão da especificação JMS na qual XMS é baseado. Esta propriedade é somente leitura.

## ***XMSC\_MAJOR\_VERSION***

**Tipo de dado:**

System.Int32

**Propriedade de:**

ConnectionMetaData

O número da versão do cliente do XMS Esta propriedade é somente leitura.

## ***XMSC\_MINOR\_VERSION***

**Tipo de dado:**

System.Int32

**Propriedade de:**

ConnectionMetaData

O número da liberação do cliente do XMS Esta propriedade é somente leitura.

## ***XMSC\_PASSWORD***

**Tipo de dado:**


matriz de byte


**Propriedade de:**

ConnectionFactory

Uma senha que pode ser usada para autenticar o aplicativo quando ele tenta se conectar a um servidor de mensagens. A senha é usada com a propriedade [XMSC\\_USERID](#)

Por padrão, a propriedade não é configurada

 Se você estiver se conectando ao IBM MQ on [Multiplataformase](#) configurar a propriedade XMSC\_USERID do connection factory, ele deverá corresponder ao **userid** do usuário conectado. Se você não configurar essas propriedades, o gerenciador de filas usará o **userid** do usuário conectado por padrão. Se você precisar de autenticação no nível de conexão adicional de usuários individuais, poderá gravar uma saída de autenticação de cliente que está configurada em IBM MQ Para obter mais informações sobre como criar uma saída de autenticação de cliente, consulte [Planejando autenticação para um aplicativo cliente](#).

 Para autenticar o usuário ao conectar ao IBM MQ for z/OS , é necessário usar uma saída de segurança.

## ***XMSC\_PRIORITY***

**Tipo de dado:**

System.Int32

**Propriedade de:**

Destino

**Nome usado em um URI:**

priority

A prioridade das mensagens enviadas para o destino.

Os valores válidos da propriedade são os seguintes:

**Valor válido**

Um número inteiro no intervalo 0, a prioridade mais baixa, para 9, a prioridade mais alta

**Significado**

Uma mensagem enviada para o destino tem a prioridade especificada A prioridade padrão do produtor de mensagens, ou qualquer prioridade especificada na chamada Enviar, é ignorada. Se o destino for uma fila IBM MQ , o valor do atributo da fila **DefPriority** também será ignorado.

<b>Valor válido</b>	<b>Significado</b>
XMSC_PRIORITY_AS_APP	Uma mensagem enviada para o destino tem a prioridade especificada na chamada Enviar. Se a chamada Enviar especificar nenhuma prioridade, a prioridade padrão do produtor da mensagem será usada em seu lugar. Se o destino for uma fila IBM MQ , o valor do atributo da fila <b>DefPriority</b> será ignorado.
XMSC_PRIORITY_AS_DEST	Se o destino for uma fila IBM MQ , uma mensagem colocada na fila terá a prioridade especificada pelo valor do atributo da fila <b>DefPriority</b> . A prioridade padrão do produtor de mensagens, ou qualquer prioridade especificada na chamada Enviar, é ignorada.  Se o destino não for uma fila IBM MQ , o significado será igual ao de XMSC_PRIORITY_AS_APP.

O valor-padrão é XMSC\_PRIORITY\_AS\_APP.

WebSphere MQ Real-Time Transport e WebSphere MQ Multicast Transport não tomam nenhuma ação com base na prioridade de uma mensagem.

### ***XMSC\_PROVIDER\_NAME***

**Tipo de dado:**

Sequência

**Propriedade de:**

ConnectionMetaData

O provedor do cliente XMS . Esta propriedade é somente leitura.

### ***XMSC\_RTT\_BROKER\_PING\_INTERVAL***

**Tipo de dado:**

System.Int32

**Propriedade de:**

ConnectionFactory

O intervalo de tempo, em milissegundos, após o qual o XMS .NET verifica a conexão com um servidor do sistema de mensagens Real Time para detectar qualquer atividade. Se nenhuma atividade for detectada, o cliente iniciará um ping; a conexão será encerrada se nenhuma resposta for detectada para o ping

O valor-padrão da propriedade é 30000.

### ***XMSC\_RTT\_CONNECTION\_PROTOCOL***

**Tipo de dado:**

System.Int32

**Propriedade de:**

ConnectionFactory

O protocolo de comunicações usado para uma conexão em tempo real com um broker.

O valor da propriedade deve ser XMSC\_RTT\_CP\_TCP, o que significa uma conexão em tempo real com um broker sobre TCP/IP. O valor padrão é XMSC\_RTT\_CP\_TCP.

### ***XMSC\_RTT\_HOST\_NAME***

**Tipo de dado:**

Sequência

**Propriedade de:**

ConnectionFactory

O nome do host ou o endereço IP do sistema no qual um broker é executado.

Essa propriedade é usada com a propriedade `XMSC_RTT_PORT` para identificar o broker

Por padrão, a propriedade não é configurada

### ***XMSC\_RTT\_LOCAL\_ADDRESS***

**Tipo de dado:**

Sequência

**Propriedade de:**

ConnectionFactory

O nome do host ou o endereço IP da interface de rede local a ser usada para uma conexão em tempo real com um broker.

Esta propriedade é útil apenas se o sistema no qual o aplicativo está em execução tiver duas ou mais interfaces de rede e você precisar ser capaz de especificar qual interface deve ser usada para uma conexão em tempo real. Se o sistema tiver apenas uma interface de rede, somente essa interface poderá ser usada. Se o sistema tiver duas ou mais interfaces de rede, e a propriedade não estiver configurada, a interface será selecionada aleatoriamente.

Por padrão, a propriedade não é configurada

### ***XMSC\_RTT\_MULTICAST***

**Tipo de dado:**

System.Int32

**Propriedade de:**

ConnectionFactory e Destino

**Nome usado em um URI:**

multicast

A configuração multicast para um connection factory ou destino. Somente um destino que é um tópico pode ter essa propriedade.

Um aplicativo usa esta propriedade para ativar multicast em associação com uma conexão em tempo real com um broker e, se multicast estiver ativado, para especificar a forma precisa na qual multicast é usado para entregar mensagens do broker para um consumidor de mensagem.. A propriedade não tem efeito sobre como um produtor de mensagem envia mensagens para o broker.

Os valores válidos da propriedade são os seguintes:

**Valor válido**

`XMSC_RTT_MULTICAST_DISABLED`

`XMSC_RTT_MULTICAST_ASCF`

**Significado**

As mensagens não são entregues a um consumidor de mensagens usando WebSphere MQ Multicast Transport. Esse valor é o valor padrão para um objeto ConnectionFactory ..

As mensagens são entregues a um consumidor de mensagens de acordo com a configuração de multicast para o connection factory associado ao consumidor de mensagem. A configuração de multicast para o connection factory é observada no momento em que a conexão é criada. Esse valor é válido apenas para um objeto de Destino e é o valor padrão para um objeto de Destino..

**Valor válido**

XMSC\_RTT\_MULTICAST\_ENABLED

**Significado**

Se o tópico for configurado para multicast no broker, as mensagens serão entregues a um consumidor de mensagem usando WebSphere MQ Multicast Transport. Uma qualidade de serviço confiável será usada, se o tópico for configurado para multicast confiável.

XMSC\_RTT\_MULTICAST\_RELIABLE

Se o tópico for configurado para multicast confiável no broker, as mensagens serão entregues a um consumidor de mensagens usando o WebSphere MQ Multicast Transport com uma qualidade de serviço confiável. Se o tópico não estiver configurado para multicast confiável, não será possível criar um consumidor de mensagens para o tópico.

XMSC\_RTT\_MULTICAST\_NOT\_RELIABLE

Se o tópico for configurado para multicast no broker, as mensagens serão entregues a um consumidor de mensagem usando WebSphere MQ Multicast Transport. Uma qualidade de serviço confiável não será usada, mesmo se o tópico estiver configurado para multicast confiável.

***XMSC\_RTT\_PORT*****Tipo de dado:**

System.Int32

**Propriedade de:**

ConnectionFactory

O número da porta na qual um broker atende às solicitações recebidas. No broker, deve-se configurar um nó de processamento de mensagens de fluxo Real-timeInput ou Real-timeOptimized para atender nessa porta.

Essa propriedade é usada com a propriedade XMSC\_RTT\_HOST\_NAME para identificar o broker.

O valor padrão da propriedade é XMSC\_RTT\_DEFAULT\_PORT ou 1506.

***XMSC\_TIME\_TO\_LIVE*****Tipo de dado:**

System.Int32

**Propriedade de:**

Destino

**Nome usado em um URI:**

expiração (para um destino IBM MQ)

timeToLive (para um destino do provedor de sistemas de mensagens padrão do WebSphere)

O tempo de vida para mensagens enviadas para o destino.

Os valores válidos da propriedade são os seguintes:

**Valor válido**

0

**Significado**

Uma mensagem enviada ao destino nunca expira.



**Valor válido**

Um inteiro positivo

**Significado**

Uma mensagem enviada para o destino tem o tempo especificado para viver em milissegundos O tempo de vida padrão do produtor de mensagem, ou qualquer tempo de vida especificado na chamada Enviar, é ignorado.

XMSC\_TIME\_TO\_LIVE\_AS\_APP

Uma mensagem enviada para o destino tem o tempo de vida especificado na chamada Enviar. Se a chamada de Envio especificar nenhum tempo de vida, o tempo de vida padrão do produtor da mensagem será usado em seu lugar

O valor padrão é XMSC\_TIME\_TO\_LIVE\_AS\_APP.

***XMSC\_USERID*****Tipo de dado:**

Sequência

**Propriedade de:**

ConnectionFactory

Um identificador de usuário que pode ser usado para autenticar o aplicativo quando ele tenta se conectar a um servidor de mensagens. O identificador de usuário é usado com a propriedade XMSC\_PASSWORD

Por padrão, a propriedade não é configurada

**Multi** Se você estiver se conectando ao IBM MQ for Multiplatformse configurar a propriedade XMSC\_USERID do connection factory, ele deverá corresponder ao **userid** do usuário conectado. Se você não configurar essas propriedades, o gerenciador de filas usará o **userid** do usuário conectado por padrão. Se você precisar de autenticação de nível de conexão adicional de usuários individuais, será possível gravar uma saída de autenticação de cliente configurada no IBM MQ. Para obter mais informações sobre como criar uma saída de autenticação de cliente, consulte Planejando autenticação para um aplicativo cliente.

**z/OS** Para autenticar o usuário ao conectar ao IBM MQ for z/OS , é necessário usar uma saída de segurança.

***XMSC\_VERSION*****Tipo de dado:**

Sequência

**Propriedade de:**

ConnectionMetaData

O identificador de versão da cliXMSent. Essa propriedade é somente leitura.

***V 9.3.0 XMSC\_WMQ\_BALANCING\_APPLICATION\_TYPE*****Tipo de dado:**

System.Int32

**Propriedade de:**

ConnectionFactory

Os valores válidos da propriedade são os seguintes:

**Valor válido**

XMSC\_WMQ\_BALANCING\_APPLICATION\_TYPE\_SIMPLE

XMSC\_WMQ\_BALANCING\_APPLICATION\_TYPE\_REQUEST\_REPLY

**Significado**

Balanceamento simples; nenhuma regra específica é aplicada além daquelas descritas em [Rebalanceamento de aplicativo de influenciamento em clusters uniformes](#). Esse é o valor-padrão.

Balanceamento de solicitação/resposta; após cada chamada MQPUT, espera-se uma chamada MQGET correspondente para uma mensagem de resposta. O balanceamento é atrasado até que tal mensagem seja recebida ou a mensagem de solicitação EXPIRAÇÃO tenha sido excedida

Além disso, essa propriedade pode ser configurada em `client.ini`. A ordem de preferência é:

1. Propriedades configuradas no aplicativo
2. Correspondência denominada [Sub-rotina do aplicativo](#) no arquivo `mqclient.ini`
3. [Sub-rotina de padrões do aplicativo](#) no arquivo `mqclient.ini`.

**V 9.3.0****XMSC\_WMQ\_BALANCING\_OPTIONS****Tipo de dado:**

System.Int32

**Propriedade de:**

ConnectionFactory

Os valores válidos da propriedade são os seguintes:

**Valor válido**

XMSC\_WMQ\_BALANCING\_OPTIONS\_NONE

XMSC\_WMQ\_BALANCING\_OPTIONS\_IGNORE\_TRANSACTION

**Valor correspondente**

Nenhuma opção configurada. Este é o valor padrão

A configuração dessa opção permite o rebalanceamento de aplicativos mesmo durante uma transação.

Além disso, essa propriedade pode ser configurada em `client.ini`. A ordem de preferência é:

1. Propriedades configuradas no aplicativo
2. Correspondência denominada [Sub-rotina do aplicativo](#) no arquivo `mqclient.ini`
3. [Sub-rotina de padrões do aplicativo](#) no arquivo `mqclient.ini`.

**V 9.3.0****XMSC\_WMQ\_BALANCING\_TIMEOUT****Tipo de dado:**

System.Int32

**Propriedade de:**

ConnectionFactory

Os valores válidos da propriedade são os seguintes:

**Valor válido**

XMSC\_WMQ\_BALANCING\_TIMEOUT\_IMMEDIATE

**Significado**

Ocorrência de tempo limite imediato

**Valor válido**

XMSC\_WMQ\_BALANCING\_TIMEOUT\_AS\_DEFAULT

XMSC\_WMQ\_BALANCING\_TIMEOUT\_NEVER

**Significado**

O valor de tempo limite padrão do conjunto. Este é o valor padrão

Nenhum tempo limite ocorre

**Nota:** O valor fornecido deve estar entre os valores definidos ou deve-se fornecer um valor de 0 a 999999999 segundos.

Além disso, essa propriedade pode ser configurada em `client.ini`. A ordem de preferência é:

1. Propriedades configuradas no aplicativo
2. Correspondência denominada Sub-rotina do aplicativo no arquivo `mqclient.ini`
3. Sub-rotina de padrões do aplicativo no arquivo `mqclient.ini`.

***XMSC\_WMQ\_BROKER\_CONTROLQ*****Tipo de dado:**

Sequência

**Propriedade de:**

ConnectionFactory

O nome da fila de controle usada pelo broker.

O valor padrão da propriedade é `SYSTEM.BROKER.CONTROL.QUEUE`.

Essa propriedade é relevante apenas no domínio de publicação / assinatura

***XMSC\_WMQ\_BROKER\_PUBQ*****Tipo de dado:**

Sequência

**Propriedade de:**

ConnectionFactory

O nome da fila monitorada por um broker na qual os aplicativos enviam mensagens que publicam.

O valor padrão da propriedade é `SYSTEM.BROKER.DEFAULT.STREAM`.

Essa propriedade é relevante apenas no domínio de publicação / assinatura

***XMSC\_WMQ\_BROKER\_QMGR*****Tipo de dado:**

Sequência

**Propriedade de:**

ConnectionFactory

O nome do gerenciador de filas ao qual um broker está conectado.

Por padrão, a propriedade não é configurada

Essa propriedade é relevante apenas no domínio de publicação / assinatura

***XMSC\_WMQ\_BROKER\_SUBQ*****Tipo de dado:**

Sequência

**Propriedade de:**

ConnectionFactory

O nome da fila de assinantes para um consumidor de mensagens não durável.

O nome da fila de assinantes deve começar com os seguintes caracteres:

SYSTEM.JMS.ND.

Se você desejar que todos os consumidores de mensagens não duráveis compartilhem uma fila de assinantes, especifique o nome completo da fila compartilhada. Uma fila com o nome especificado deve existir antes que um aplicativo possa criar um consumidor de mensagens não duráveis.

Se desejar que cada consumidor de mensagens não duráveis recupere mensagens de sua própria fila de assinantes exclusivos, especifique um nome de fila que termine com um asterisco (\*). Em seguida, quando um aplicativo cria um consumidor de mensagens não duráveis, o cliente XMS cria uma fila dinâmica para uso exclusivo pelo consumidor de mensagens.. O cliente XMS usa o valor da propriedade para configurar o conteúdo do campo **DynamicQName** no descritor de objeto usado para criar a fila dinâmica.

O valor padrão da propriedade é SYSTEM.JMS.ND.SUBSCRIBER.QUEUE, que significa que XMS usa a abordagem de fila compartilhada por padrão.

Essa propriedade é relevante apenas no domínio de publicação / assinatura

### ***XMSC\_WMQ\_BROKER\_VERSION***

**Tipo de dado:**

System.Int32

**Propriedade de:**

ConnectionFactory e Destino

**Nome usado em um URI:**

brokerVersion

O tipo de corretor usado pelo aplicativo para uma conexão ou para o destino. Somente um destino que é um tópico pode ter essa propriedade

Os valores válidos da propriedade são os seguintes:

<b>Valor válido</b>	<b>Significado</b>
XMSC_WMQ_BROKER_V1	O aplicativo está usando um broker de publicação / assinatura do IBM MQ  O aplicativo também poderá usar esse valor se você migrar de IBM MQ publicar / assinar para WebSphere Message Broker , mas não alterou o aplicativo.
XMSC_WMQ_BROKER_V2	O aplicativo está usando um broker de IBM Integration Bus
XMSC_WMQ_BROKER_UNSPECIFIED	Após o broker ser migrado, configure essa propriedade para que os cabeçalhos RFH2 não sejam mais usados. Após a migração, essa propriedade não é mais relevante.

O valor padrão para um connectionfactory é XMSC\_WMQ\_BROKER\_UNESPECIFICADO mas, por padrão, a propriedade não é configurada para um destino. A configuração da propriedade para um destino substitui qualquer valor especificado pela propriedade do connection factory.

### ***XMSC\_WMQ\_CCDTURL***

**Tipo de dado:**

System.String

**Propriedade de:**

ConnectionFactory

**Objetos aplicáveis:**

Nome longo da ferramenta de administração JMS: CCDTURL

Nome abreviado da ferramenta de administração JMS: CCDT

Um Localizador Uniforme de Recursos (URL) que identifica o nome e o local do arquivo que contém a tabela de definição de canal do cliente e também especifica como o arquivo pode ser acessado.

Por padrão, essa propriedade não está configurada.

### ***XMSC\_WMQ\_CCSID***

**Tipo de dado:**

System.Int32

**Propriedade de:**

Destino

**Nome usado em um URI:**

CCSID

O identificador (CCSID) do conjunto de caracteres codificados, ou página de códigos, em que as cadeias de dados de caracteres no corpo de uma mensagem estão quando o cliente XMS encaminha a mensagem para o destino. Se configurado para uma mensagem individual, a propriedade JMS\_IBM\_CHARACTER\_SET substitui o CCSID especificado para o destino por essa propriedade.

O valor-padrão da propriedade é 1208.

Essa propriedade é relevante apenas para mensagens enviadas para o destino, não para mensagens recebidas do destino

### ***XMSC\_WMQ\_CHANEXO***

**Tipo de dado:**

Sequência

**Propriedade de:**

ConnectionFactory

**Objetos aplicáveis:**

Nome longo da ferramenta de administração JMS: CHANNEL

Nome abreviado da ferramenta de administração JMS: CHAN

O nome do canal a ser usado para uma conexão.

Por padrão, a propriedade não é configurada

Essa propriedade é relevante somente quando um aplicativo se conecta a um gerenciador de fila no modo de cliente

### ***XMSC\_WMQ\_CLIENT\_RECONNECT\_OPTIONS***

**Tipo de dado:**

Sequência

**Propriedade de:**

ConnectionFactory

**Objetos aplicáveis:**

Nome longo da ferramenta de administração JMS: CLIENTRECONNECTOPTIONS

Nome abreviado da ferramenta de administração JMS: CROPT

Esta propriedade especifica as opções de reconexão do cliente para novas conexões criadas por este factory. Ele é encontrado no XMSC e é um dos seguintes:

- WMQ\_CLIENT\_RECONNECT\_AS\_DEF (padrão).. Use o valor especificado no arquivo `mqclient.ini`. Configure o valor usando a propriedade **DefRecon** dentro da sub-rotina Canais. Ele pode ser configurado para um de:
  1. Sim. Comporta-se como a opção WMQ\_CLIENT\_RECONNECT
  2. NÃO. Padrão. Não especifica nenhuma opção de reconexão
  3. QMGR. Comporta-se como a opção WMQ\_CLIENT\_RECONNECT\_Q\_MGR

4. Desativar. Comporta-se como a opção WMQ\_CLIENT\_RECONNECT\_DISABLED
- WMQ\_CLIENT\_RECONNECT. Reconecte a qualquer um dos gerenciadores de filas especificados na lista de nomes de conexões.
  - WMQ\_CLIENT\_RECONNECT\_Q\_MGR. Reconecta ao mesmo gerenciador de filas ao qual ele está originalmente conectado. Ele retorna MQRC\_RECONNECT\_QMID\_MISMATCH se o gerenciador de filas ao qual ele tenta se conectar (especificado na lista de nomes de conexão) tiver um QMID diferente para o gerenciador de filas originalmente conectado.
  - WMQ\_CLIENT\_RECONNECT\_DISABLED. A reconexão está desativada

### ***XMSC\_WMQ\_CLIENT\_RECONNECT\_TIMEOUT***

**Tipo de dado:**

Sequência

**Propriedade de:**

ConnectionFactory

**Objetos aplicáveis:**

Nome longo da ferramenta de administração JMS: CLIENTRECONNECTTIMEOUT

Nome abreviado da ferramenta de administração JMS: CRT

A propriedade XMSC\_WMQ\_CLIENT\_RECONNECT\_TIMEOUT é válida apenas para o cliente XMS .NET Gerenciado

Esta propriedade especifica a duração de tempo, em segundos, que uma conexão do cliente tenta reconectar.

Depois de tentar se reconectar por esse tempo, o cliente falhará com MQRC\_RECONNECT\_FAILED. A configuração padrão para esta propriedade é XMSC.WMQ\_CLIENT\_RECONNECT\_TIMEOUT\_DEFAULT.

O valor-padrão desta propriedade é 1800.

### ***XMSC\_WMQ\_CONNECTION\_MODE***

**Tipo de dado:**

System.Int32

**Propriedade de:**

ConnectionFactory

O modo pelo qual um aplicativo se conecta a um gerenciador de filas.

Os valores válidos da propriedade são os seguintes:

<b>Valor válido</b>	<b>Significado</b>
XMSC_WMQ_CM_BINDINGS	Uma conexão com um gerenciador de fila no modo de ligações, para obter um desempenho ideal Esse valor é o valor padrão para C/C + +.
XMSC_WMQ_CM_CLIENT	Uma conexão com um gerenciador de filas no modo cliente para assegurar uma pilha totalmente gerenciada. Esse valor é o valor padrão para .NET.
XMSC_WMQ_CM_CLIENT_UNMANAGED (apenas para .NET)	Uma conexão com um gerenciador de filas que força uma pilha de clientes não gerenciada

### ***XMSC\_WMQ\_CONNECTION\_NAME\_LIST***

**Tipo de dado:**

Sequência

**Propriedade de:**

ConnectionFactory

**Objetos aplicáveis:**

Nome longo da ferramenta de administração JMS: CONNECTIONNAMESLIST

Nome abreviado da ferramenta de administração JMS: CNLIST

Esta propriedade especifica os hosts aos quais o cliente tenta se reconectar depois que sua conexão é interrompida

A lista de nomes de conexão é uma lista separada por vírgula de pares de portas host / ip. A configuração padrão para essa propriedade é WMQ\_CONNECTION\_NAME\_LIST\_DEFAULT

Por exemplo,127.0.0.1 (1414) ,host2.example.com(1400)

A configuração padrão desta propriedade é localhost (1414).

***XMSC\_WMQ\_DUR\_SUBQ*****Tipo de dado:**

Sequência

**Propriedade de:**

Destino

O nome da fila de assinantes para um assinante durável que está recebendo mensagens do destino. Somente um destino que é um tópico pode ter essa propriedade

O nome da fila de assinantes deve começar com os seguintes caracteres:

SYSTEM.JMS.D.

Se você deseja que todos os assinantes duráveis compartilhem uma fila de assinantes, especifique o nome completo da fila compartilhada. Uma fila com o nome especificado deve existir antes que um aplicativo possa criar um assinante durável

Se deseja que cada assinante durável recupere mensagens de sua própria fila de assinantes exclusivos, especifique um nome da fila que termine com um asterisco (\*). Então, quando um aplicativo cria um assinante durável, o cliente XMS cria uma fila dinâmica para uso exclusivo pelo assinante durável. O cliente XMS usa o valor da propriedade para configurar o conteúdo do campo **DynamicQName** no descritor de objeto usado para criar a fila dinâmica.

O valor padrão da propriedade é SYSTEM.JMS.D.SUBSCRIBER.QUEUE, que significa que XMS usa a abordagem de fila compartilhada por padrão.

Essa propriedade é relevante apenas no domínio de publicação / assinatura

***XMSC\_WMQ\_ENCODING*****Tipo de dado:**

System.Int32

**Propriedade de:**

Destino

Como os dados numéricos no corpo de uma mensagem são representados quando o cliente XMS encaminha a mensagem para o destino. Se configurado para uma mensagem individual, a propriedade JMS\_IBM\_ENCODING substituirá a codificação especificada para o destino por esta propriedade. A propriedade especifica a representação de números inteiros binários, números inteiros decimais empacotados e números de vírgula flutuante..

Os valores válidos da propriedade são iguais aos valores que podem ser especificados no campo **Encoding** de um descritor de mensagens.

Um aplicativo pode usar as seguintes constantes nomeadas para configurar a propriedade:

**Constante nomeada**

MQENC\_INTEGER\_NORMAL

MQENC\_INTEGER\_REVERSED

**Significado**

Codificação de número inteiro normal

Codificação de número inteiro revertida

<b>Constante nomeada</b>	<b>Significado</b>
MQENC_DECIMAL_NORMAL	Codificação decimal compactada normal
MQENC_DECIMAL_REVERSED	Codificação decimal compactada revertida
MQENC_FLOAT_IEEE_NORMAL	Codificação de ponto flutuante IEEE normal
MQENC_FLOAT_IEEE_REVERSED	Codificação de ponto flutuante IEEE revertida
MQENC_FLOAT_S390	Codificação de ponto flutuante da arquitetura do z/OS
MQENC_NATIVE	Codificação de máquina nativa

Para formar um valor para a propriedade, o aplicativo pode incluir três dessas constantes da seguinte forma:

- Uma constante cujo nome começa com MQENC\_INTEGER, para especificar a representação de inteiros binários
- Uma constante cujo nome começa com MQENC\_DECIMAL, para especificar a representação de números inteiros decimais compactados
- Uma constante cujo nome começa com MQENC\_FLOAT, para especificar a representação de números de ponto flutuante

Como alternativa, o aplicativo pode configurar a propriedade para MQENC\_NATIVE, cujo valor é dependente de ambiente.

O valor padrão da propriedade é MQENC\_NATIVE.

Essa propriedade é relevante apenas para mensagens enviadas para o destino, não para mensagens recebidas do destino

## ***XMSC\_WMQ\_FAIL\_IF\_QUIESCE***

### **Tipo de dado:**

System.Int32

### **Propriedade de:**

ConnectionFactory e Destino

### **Nome usado em um URI:**

failIfQuiesce

### **Objetos aplicáveis:**

Nome longo da ferramenta de administração JMS: FAILIFQUIESCE

Nome abreviado da ferramenta de administração JMS: FIQ

Se as chamadas para determinados métodos falharem se o gerenciador de filas ao qual o aplicativo está conectado estiver em um estado quiesce.

Os valores válidos da propriedade são os seguintes:

<b>Valor válido</b>	<b>Significado</b>
XMSC_WMQ_FIQ_YES	Chamadas para determinados métodos falharão se o gerenciador de filas estiver em um estado quiesce. Quando o aplicativo detectar que o gerenciador de filas está em quiesce, o aplicativo poderá concluir sua tarefa imediata e fechar a conexão, permitindo que o gerenciador de filas pare.
XMSC_WMQ_FIQ_NO	Nenhuma chamada de método falha porque o gerenciador de fila está em um estado de quiesce. Se você especificar esse valor, o aplicativo não poderá detectar que o gerenciador de filas está quiesce. O aplicativo pode continuar a executar operações no gerenciador de fila e, portanto, evitar que o gerenciador de filas pare.



O valor padrão para um connection factory é XMSC\_WMQ\_FIQ\_YES mas, por padrão, a propriedade não é configurada para um destino. A configuração da propriedade para um destino substitui qualquer valor especificado pela propriedade do connection factory.

### ***XMSC\_WMQ\_MESSAGE\_BODY***

**Tipo de dado:**

System.Int32

**Propriedade de:**

Destino

Essa propriedade determina se um aplicativo XMS processa o MQRFH2 de uma mensagem IBM MQ como parte da carga útil da mensagem (ou seja, como parte do corpo da mensagem).

**Nota:** Ao enviar mensagens para um destino, a propriedade XMSC\_WMQ\_MESSAGE\_BODY substitui a propriedade de destino XMSC\_WMQ\_TARGET\_CLIENT existente do XMS

Os valores válidos para essa propriedade são:

### ***XMSC\_WMQ\_MESSAGE\_BODY\_JMS***

**Receber:** o tipo e o corpo da mensagem de XMS entrada são determinados pelo conteúdo do MQRFH2 (se presente) ou do MQMD (se não houver MQRFH2) na mensagem IBM MQ recebida.

**Enviar:** O corpo da mensagem XMS de saída contém um cabeçalho MQRFH2 pré-anexado e gerado automaticamente com base nas propriedades da Mensagem e nos campos do cabeçalho XMS .

### ***XMSC\_WMQ\_MESSAGE\_BODY\_MQ***

**Receber:** O tipo de mensagem de XMS entrada é sempre ByteMessage. Indiferente do conteúdo da mensagem IBM MQ recebida ou do campo de formato do MQMD recebido. O corpo da mensagem XMS é os dados da mensagem inalterados retornados pela chamada API do provedor de sistemas de mensagens subjacente. O conjunto de caracteres e a codificação dos dados no corpo da mensagem são determinados pelos campos CodedCharSetId e Codificação do MQMD. O formato dos dados no corpo da mensagem é determinado pelo campo Formato do MQMD.

**Enviar:** o corpo da mensagem do XMS de saída contém a carga útil do aplicativo como está; e nenhum cabeçalho IBM MQ gerado automaticamente é incluído no corpo.

### ***XMSC\_WMQ\_MESSAGE\_BODY\_UNSPECIFIED***

**Receber:** O cliente XMS determina um valor adequado para esta propriedade.. No caminho de recebimento, esse valor é o valor da propriedade WMQ\_MESSAGE\_BODY\_JMS.

**Enviar:** O cliente XMS determina um valor adequado para essa propriedade. No caminho de envio, esse valor é o valor da propriedade XMSC\_WMQ\_TARGET\_CLIENT.

Por padrão, essa propriedade é configurada como XMSC\_WMQ\_MESSAGE\_BODY\_UNSPECIFIED

### ***XMSC\_WMQ\_MQMD\_MESSAGE\_CONTEXT***

**Tipo de dado:**

System.Int32

**Propriedade de:**

Destino

Determina qual nível de contexto de mensagem deve ser configurado pelo aplicativo XMS . O aplicativo deve estar em execução com autoridade de contexto apropriado para esta propriedade entrar em vigor.

Os valores válidos para essa propriedade são:

### ***XMSC\_WMQ\_MDCTX\_DEFAULT***

Para mensagens de saída, a chamada da API MQOPEN e a estrutura MQPMO não especificam opções de contexto de mensagem explícitas

### **XMSC\_WMQ\_MDCTX\_SET\_IDENTITY\_CONTEXT**

A chamada da API MQOPEN especifica a opção de contexto da mensagem MQOO\_SET\_IDENTITY\_CONTEXT e a estrutura MQPMO especifica MQPMO\_SET\_IDENTITY\_CONTEXT

### **XMSC\_WMQ\_MDCTX\_SET\_ALL\_CONTEXT**

A chamada da API MQOPEN especifica a opção de contexto da mensagem MQOO\_SET\_ALL\_CONTEXT e a estrutura MQPMO especifica MQPMO\_SET\_ALL\_CONTEXT

Por padrão, essa propriedade é configurada para XMSC\_WMQ\_MDCTX\_DEFAULT

**Nota:** Essa propriedade não é relevante quando um aplicativo se conecta ao WebSphere Application Server service integration bus

As propriedades a seguir requerem que a propriedade XMSC\_WMQ\_MQMD\_MESSAGE\_CONTEXT seja configurada como o valor da propriedade XMSC\_WMQ\_MDCTX\_SET\_IDENTITY\_CONTEXT ou o valor da propriedade XMSC\_WMQ\_MDCTX\_SET\_ALL\_CONTEXT ao enviar uma mensagem para que tenha efeito desejado:

- JMS\_IBM\_MQMD\_UserIdentifier
- JMS\_IBM\_MQMD\_AccountingToken
- JMS\_IBM\_MQMD\_ApplIdentityData

As propriedades a seguir requerem que a propriedade XMSC\_WMQ\_MQMD\_MESSAGE\_CONTEXT seja configurada para o valor da propriedade XMSC\_WMQ\_MDCTX\_SET\_ALL\_CONTEXT ao enviar uma mensagem para que tenha efeito desejado:

- JMS\_IBM\_MQMD\_PutApplType
- JMS\_IBM\_MQMD\_PutApplName
- JMS\_IBM\_MQMD\_PutDate
- JMS\_IBM\_MQMD\_PutTime
- JMS\_IBM\_MQMD\_ApplOriginData

### **XMSC\_WMQ\_MQMD\_READ\_ENABLED**

#### **Tipo de dado:**

System.Int32

#### **Propriedade de:**

Destino

Essa propriedade determina se um aplicativo XMS pode ou não extrair os valores dos campos MQMD

Os valores válidos para essa propriedade são:

#### **XMSC\_WMQ\_READ\_ENABLED\_NO**

Ao enviar mensagens, as propriedades JMS\_IBM\_MQMD\* em uma mensagem enviada não são atualizadas para refletir os valores de campos atualizados no MQMD.

Ao receber mensagens, nenhuma das propriedades JMS\_IBM\_MQMD\* está disponível em uma mensagem recebida, mesmo que algumas ou todas elas sejam configuradas pelo emissor.

#### **XMSC\_WMQ\_READ\_ENABLED\_YES**

Ao enviar mensagens, todas as propriedades JMS\_IBM\_MQMD\* em uma mensagem enviada são atualizadas para refletir os valores de campo atualizados no MQMD, incluindo as propriedades que o emissor não configurou explicitamente...

Ao receber mensagens, todas as propriedades JMS\_IBM\_MQMD\* estão disponíveis em uma mensagem recebida, incluindo aquelas propriedades que o emissor não configurou explicitamente.

Por padrão, essa propriedade é configurada como XMSC\_WMQ\_READ\_ENABLED\_NO

### **XMSC\_WMQ\_MQMD\_WRITE\_ENABLED**

#### **Tipo de dado:**

System.Int32

**Propriedade de:**

Destino

Essa propriedade determina se um aplicativo XMS pode ou não configurar os valores dos campos MQMD

Os valores válidos para essa propriedade são:

**XMSC\_WMQ\_WRITE\_ENABLED\_NO**

Todas as propriedades JMS\_IBM\_MQMD\* são ignoradas e seus valores não são copiados para a estrutura MQMD subjacente.

**XMSC\_WMQ\_WRITE\_ENABLED\_YES**

As propriedades JMS\_IBM\_MQMD\* são processadas. Seus valores serão copiados para a estrutura do MQMD subjacente.

Por padrão, essa propriedade é configurada como XMSC\_WMQ\_WRITE\_ENABLED\_NO

***XMSC\_WMQ\_PUT\_ASYNC\_ALLOWED*****Tipo de dado:**

System.Int32

**Propriedade de:**

Destino

Essa propriedade determina se os produtores de mensagens têm permissão para usar as postagens assíncronas para enviar mensagens para esse destino.

Os valores válidos para essa propriedade são:

**XMSC\_WMQ\_PUT\_ASYNC\_ALLOWED\_AS\_DEST**

Determine se as colocações assíncronas são permitidas consultando a definição de fila ou tópico.

**XMSC\_WMQ\_PUT\_ASYNC\_ALLOWED\_AS\_Q\_DEF**

Determine se as entradas assíncronas são permitidas consultando a definição de fila.

**XMSC\_WMQ\_PUT\_ASYNC\_ALLOWED\_AS\_TOPIC\_DEF**

Determine se as colocações assíncronas são permitidas referindo-se à definição de tópico

**XMSC\_WMQ\_PUT\_ASYNC\_ALLOWED\_DISABLED**

Puts assíncronos não são permitidos.

**XMSC\_WMQ\_PUT\_ASYNC\_ALLOWED\_ENABLED**

As colocações assíncronas são permitidas

Por padrão, essa propriedade é configurada como XMSC\_WMQ\_PUT\_ASYNC\_ALLOWED\_AS\_DEST

**Nota:** Essa propriedade não é relevante quando um aplicativo está se conectando ao WebSphere Application Server service integration bus

***XMSC\_WMQ\_READ\_AHEAD\_ALLOWED*****Tipo de dado:**

System.Int32

**Propriedade de:**

Destino

Essa propriedade determina se os consumidores de mensagens e os navegadores de fila têm permissão para usar leitura antecipada para obter mensagens não persistentes, não transacionais desse destino em um buffer interno antes de recebê-las.

Os valores válidos para essa propriedade são:

**XMSC\_WMQ\_READ\_AHEAD\_ALLOWED\_AS\_Q\_DEF**

Determine se a leitura antecipada é permitida referindo-se à definição de fila

**XMSC\_WMQ\_READ\_AHEAD\_ALLOWED\_AS\_TÓPICO\_DEF**

Determine se a leitura antecipada é permitida referindo-se à definição de tópico

**XMSC\_WMQ\_READ\_AHEAD\_ALLOWED\_AS\_DEST**

Determine se a leitura antecipada é permitida consultando a definição de fila ou tópico.

**XMSC\_WMQ\_READ\_AHEAD\_ALLOWED\_DISABLED**

A leitura antecipada não é permitida ao consumir ou procurar mensagens.

**XMSC\_WMQ\_READ\_AHEAD\_ALLOWED\_ENABLED**

A leitura antecipada é permitida

Por padrão, essa propriedade é configurada como XMSC\_WMQ\_READ\_AHEAD\_ALLOWED\_AS\_DEST

***XMSC\_WMQ\_READ\_AHEAD\_CLOSE\_POLICY*****Tipo de dado:**

System.Int32

**Propriedade de:**

Destino

Para mensagens que estão sendo entregues em um listener de mensagem assíncrona, essa propriedade determina o que acontece com as mensagens no buffer de leitura antecipada interno quando o consumidor de mensagens é fechado.

Essa propriedade é aplicável ao especificar opções de fila de fechamento ao consumir mensagens de um destino e não aplicável ao enviar mensagens para um destino.

Essa propriedade é ignorada para navegadores de fila, pois durante a procura as mensagens ainda estão disponíveis nas filas.

Os valores válidos para essa propriedade são:

**XMSC\_WMQ\_READ\_AHEAD\_CLOSE\_POLICY\_DELIVER\_CURRENT**

Apenas a chamada do listener de mensagem atual é concluída antes de retornar, potencialmente deixando mensagens no buffer de leitura antecipada interno, que são, então, descartados

**XMSC\_WMQ\_READ\_AHEAD\_CLOSE\_POLICY\_DELIVER\_ALL**

Todas as mensagens no buffer de leitura antecipada interno são entregues para o listener de mensagens do aplicativo antes de retornar.

Por padrão, essa propriedade é configurada como XMSC\_WMQ\_READ\_AHEAD\_CLOSE\_POLICY\_DELIVER\_CURRENT

**Nota:****Terminação anormal do aplicativo**

Todas as mensagens no buffer de leitura antecipada são perdidas quando um aplicativo XMS é finalizado abruptamente.

**Implicações para transações**

A leitura antecipada é desativada quando os aplicativos usam transações.. Portanto, o aplicativo não está vendo nenhuma diferença no comportamento quando eles usam sessões transacionados

**Implicações dos modos de confirmação de sessão**

A leitura antecipada é ativada em uma sessão não transacionada quando os modos de confirmação são XMSC\_AUTO\_RECONHEÇO ou XMSC\_DUPS\_OK\_RECONHEÇO A leitura antecipada será desativada se o modo de confirmação de sessão for XMSC\_CLIENT\_RECONHEÇO, independentemente de sessões transacionadas ou não transacionados

### **Implicações para navegadores de filas e seletores de navegador de fila**

Os navegadores de fila e os seletores de navegador de filas, usados em aplicativos XMS , obtêm a vantagem de desempenho da leitura antecipada de. Fechar o navegador de filas não compromete o desempenho porque a mensagem ainda está disponível na fila para operações adicionais. Não há outras implicações para navegadores de filas e seletores de navegador de filas além dos benefícios de desempenho de leitura antecipada.

### ***XMSC\_WMQ\_HOST\_NAME***

**Tipo de dado:**

Sequência

**Propriedade de:**

ConnectionFactory

**Objetos aplicáveis:**

Nome longo da ferramenta de administração JMS: HOSTNAME

Nome abreviado da ferramenta de administração JMS: HOST

O nome do host ou o endereço IP do sistema no qual um gerenciador de filas é executado.

Essa propriedade é usada apenas quando um aplicativo se conecta a um gerenciador de filas em modo cliente A propriedade é utilizada com a propriedade XMSC\_WMQ\_PORT para identificar o gerenciador de filas.

O valor padrão da propriedade é localhost..

### ***XMSC\_WMQ\_LOCAL\_ADDRESS***

**Tipo de dado:**

Sequência

**Propriedade de:**

ConnectionFactory

**Objetos aplicáveis:**

Nome longo da ferramenta de administração JMS: LOCALADDRESS

Nome abreviado da ferramenta de administração JMS: LA

Para uma conexão com um gerenciador de filas, essa propriedade especifica a interface de rede local a ser usada, a porta local ou o intervalo de portas locais a serem usadas ou ambos.

O valor da propriedade é uma sequência com o seguinte formato:

*[host\_name] [(low\_port) [,high\_port]]*

Os significados das variáveis são os seguintes:

***host\_name***

O nome do host ou o endereço IP da interface de rede local a ser usado para a conexão

Fornecer essas informações é necessário apenas se o sistema no qual o aplicativo está em execução tiver duas ou mais interfaces de rede e você precisar ser capaz de especificar qual interface deve ser usada para a conexão.. Se o sistema tiver apenas uma interface de rede, somente essa interface poderá ser usada Se o sistema tiver duas ou mais interfaces de rede e você não especificar qual interface deve ser usada, a interface será selecionada aleatoriamente

***low\_port***

O número da porta local a ser usada para a conexão

Se *high\_port* também for especificado, *low\_port* será interpretado como o número da porta mais baixo em um intervalo de números de portas

***high\_port***

O número da porta mais alto em um intervalo de números de porta Uma das portas no intervalo especificado deve ser usada para a conexão..

O comprimento máximo da sequência é de 48 caracteres.

Aqui estão alguns exemplos de valores válidos da propriedade:

JÚPITER  
9.20.4.98  
JUPITER (1000)  
9.20.4.98(1000,2000)  
(1000)  
(1000,2000)

Por padrão, a propriedade não é configurada

Essa propriedade é relevante somente quando um aplicativo se conecta a um gerenciador de fila no modo de cliente

### ***XMSC\_WMQ\_MESSAGE\_SELECTION***

**Tipo de dado:**

System.Int32

**Propriedade de:**

ConnectionFactory

Determina se a seleção de mensagens é feita por XMS cliente ou pelo broker.

Os valores válidos da propriedade são os seguintes:

<b>Valor válido</b>	<b>Significado</b>
XMSC_WMQ_MSEL_CLIENT	A seleção de mensagens é feita pelo cliente do XMS
XMSC_WMQ_MSEL_BROKER	A seleção de mensagens é feita pelo broker

O valor padrão é XMSC\_WMQ\_MSEL\_CLIENT.

Essa propriedade é relevante apenas no domínio de publicação / assinatura. A seleção de mensagens pelo broker não será suportada se a propriedade XMSC\_WMQ\_BROKER\_VERSION estiver configurada como XMSC\_WMQ\_BROKER\_V1.

### ***XMSC\_WMQ\_MSG\_BATCH\_SIZE***

**Tipo de dado:**

System.Int32

**Propriedade de:**

ConnectionFactory

O número máximo de mensagens a serem recuperadas de uma fila em um lote ao usar a entrega de mensagem assíncrona.

Quando um aplicativo está usando a entrega de mensagem assíncrona, sob determinadas condições, o cliente XMS recupera um lote de mensagens de uma fila antes de encaminhar cada mensagem individualmente para o aplicativo. Esta propriedade especifica o número máximo de mensagens que podem estar no lote.

O valor da propriedade é um inteiro positivo e o valor padrão é 10. Considere configurar a propriedade para um valor diferente somente se você tiver um problema de desempenho específico que precisa ser abordado.

Se um aplicativo for conectado a um gerenciador de filas por meio de uma rede, aumentar o valor dessa propriedade pode reduzir sobrecargas de rede e tempos de resposta, mas aumentar a quantidade de memória necessária para armazenar as mensagens no sistema do cliente. Por outro lado, reduzir o valor dessa propriedade pode aumentar as sobrecargas de rede e os tempos de resposta, mas reduzir a quantidade de memória necessária para armazenar as mensagens..

## ***XMSC\_WMQ\_POLLING\_INTERVAL***

**Tipo de dado:**

System.Int32

**Propriedade de:**

ConnectionFactory

Se cada listener de mensagem dentro de uma sessão não tiver mensagens adequadas em sua fila, este valor será o intervalo máximo, em milissegundos, que decorrerá antes que cada listener da mensagem tente novamente obter uma mensagem de sua fila.

Se ocorrer com frequência o fato de nenhuma mensagem adequada estar disponível para qualquer um dos listeners da mensagem em uma sessão, considere aumentar o valor desta propriedade.

O valor da propriedade é um número inteiro positivo. O valor-padrão é 5000.

## ***XMSC\_WMQ\_PORT***

**Tipo de dado:**

System.Int32

**Propriedade de:**

ConnectionFactory

**Objetos aplicáveis:**

Nome longo da ferramenta de administração JMS: PORT

Nome abreviado da ferramenta de administração JMS: PORT

O número da porta na qual um gerenciador de filas atende às solicitações recebidas.

Essa propriedade é usada apenas quando um aplicativo se conecta a um gerenciador de filas em modo cliente. A propriedade é usada com a propriedade XMSC\_WMQ\_HOST\_NAME para identificar o gerenciador de filas.

O valor padrão da propriedade é XMSC\_WMQ\_DEFAULT\_CLIENT\_PORT ou 1414.

## ***XMSC\_WMQ\_PROVIDER\_VERSION***

**Tipo de dado:**

Sequência

**Propriedade de:**

ConnectionFactory

A versão, liberação, nível de modificação e fix pack do gerenciador de filas ao qual o aplicativo pretende se conectar. Valores válidos para esta propriedade são:

- Não especificado

Ou uma sequência em um dos seguintes formatos

- V.R.M.F
- V.R.M
- V.R
- V

em que V, R, M e F são valores inteiros maiores ou iguais a zero.

Por padrão, essa propriedade é configurada como "não especificada"..

A versão do IBM MQ Client também desempenha uma função principal se um aplicativo cliente XMS pode usar recursos específicos do IBM MQ.

**Nota:** Uma propriedade de sistema XMSC\_WMQ\_OVERRIDEPROVIDERVERSION substitui a propriedade XMSC\_WMQ\_PROVIDER\_VERSION.. Esta propriedade pode ser usada se você não conseguir alterar a configuração do connection factory

## ***XMSC\_WMQ\_PUB\_ACK\_INTERVAL***

**Tipo de dado:**

System.Int32

**Propriedade de:**

ConnectionFactory

O número de mensagens publicadas por um publicador antes do cliente XMS solicitar uma confirmação do broker.

Se você diminuir o valor dessa propriedade, o cliente solicita confirmações com mais frequência e, portanto, o desempenho do publicador diminui. Se aumentar o valor, o cliente terá mais tempo para emitir uma exceção se o intermediário falhar.

O valor da propriedade é um número inteiro positivo. O valor padrão é 25.

## ***XMSC\_WMQ\_QMGR\_CCSID***

**Tipo de dado:**

System.Int32

**Propriedade de:**

ConnectionFactory

O identificador (CCSID) do conjunto de caractere codificado, ou página de códigos, no qual os campos de dados de caractere definidos na Message Queue Interface (MQI) são trocados entre o cliente XMS e o cliente IBM MQ . Essa propriedade não se aplica às cadeias de dados de caracteres nos corpos das mensagens

Quando o aplicativo XMS se conecta a um gerenciador de fila no modo cliente, o cliente XMS vincula ao cliente IBM MQ . As informações trocadas entre os dois clientes contêm campos de dados de caracteres definidos no MQI. Sob circunstâncias normais, o cliente IBM MQ assume que esses campos estão na página de códigos do sistema no qual os clientes estão em execução. Se o cliente XMS fornecer e esperar receber esses campos em uma página de código diferente, deve-se configurar essa propriedade para informar o cliente IBM MQ

Quando o cliente IBM MQ encaminha esses campos de dados de caractere para o gerenciador de filas, os dados neles devem ser convertidos, se necessário, na página de códigos usada pelo gerenciador de fila. Da mesma forma, quando o cliente IBM MQ recebe esses campos do gerenciador de filas, os dados neles devem ser convertidos, se necessário, para a página de código na qual o cliente XMS espera receber os dados. O cliente IBM MQ usa essa propriedade para executar essa conversão de dados.

Por padrão, a propriedade não é configurada

A configuração dessa propriedade é equivalente à configuração da variável de ambiente MQCCSID para um cliente IBM MQ que suporta aplicativos clientes nativos IBM MQ . Para obter mais informações sobre esta variável de ambiente, consulte [MQCCSID](#)

## ***XMSC\_WMQ\_QUEUE\_MANAGER***

**Tipo de dado:**

Sequência

**Propriedade de:**

ConnectionFactory

**Objetos aplicáveis:**

Nome longo da ferramenta de administração JMS: QMANAGER

Nome abreviado da ferramenta de administração JMS: QMGR

O nome do gerenciador de filas para conexão.

Por padrão, a propriedade não é configurada



## ***XMSC\_WMQ\_RECEIVE\_CCSID***

A propriedade de destino que configura o destino CCSID para a conversão de mensagens do gerenciador de filas. O valor é ignorado a menos que XMSC\_WMQ\_RECEIVE\_CONVERSION seja configurado como WMQ\_RECEIVE\_CONVERSION\_QMGR.

### **Tipo de dado:**

Integer

### **Valor:**

Qualquer número inteiro positivo.

O valor padrão é 1208.

Especificar um valor GMO\_CONVERT em uma mensagem é opcional. Se um valor GMO\_CONVERT for especificado, a conversão ocorrerá de acordo com o valor especificado.

## ***XMSC\_WMQ\_RECEIVE\_CONVERSION***

A propriedade de destino que determina se a conversão de dados será executada pelo gerenciador de filas.

### **Tipo de dado:**

Integer

### **Valores:**

XMSC\_WMQ\_RECEIVE\_CONVERSION\_CLIENT\_MSG (DEFAULT): Execute a conversão de dados somente no cliente XMS A conversão é sempre feita usando a página de códigos 1208

XMSC\_WMQ\_RECEIVE\_CONVERSION\_QMGR: Executar conversão de dados no gerenciador de filas antes de enviar uma mensagem para o cliente XMS .

## ***XMSC\_WMQ\_RECEIVE\_EXIT***

### **Tipo de dado:**

Sequência

### **Propriedade de:**

ConnectionFactory

Identifica uma saída de recebimento do canal para ser executada.

O valor da propriedade é uma sequência que identifica uma saída de recebimento do canal e possui o seguinte formato:

**libraryName**(entryPointNome)

em que,

- **libraryName** é o caminho completo da saída gerenciada .dll
- entryPointName é o nome da classe qualificado pelo namespace.

Por exemplo, C:\MyReceiveExit.dll(MyReceiveExitNameSpace.MyReceiveExitClassName)

Por padrão, a propriedade não é configurada

Essa propriedade é relevante somente quando um aplicativo se conecta a um gerenciador de filas no modo de cliente gerenciado Além disso, apenas as saídas gerenciadas são suportadas

## ***XMSC\_WMQ\_RECEIVE\_EXIT\_INIT***

### **Tipo de dado:**

Sequência

### **Propriedade de:**

ConnectionFactory

Os dados do usuário que são transmitidos para uma saída de recebimento de canal quando ela é chamada.

O valor da propriedade é uma sequência. Por padrão, a propriedade não é configurada

Essa propriedade é relevante apenas quando um aplicativo se conecta a um gerenciador de filas no modo de cliente gerenciado e a propriedade “XMSC\_WMQ\_RECEIVE\_EXIT” na página 2149 é configurada

### ***XMSC\_WMQ\_RESOLVED\_QUEUE\_MANAGER***

**Tipo de dado:**

Sequência

**Propriedade de:**

ConnectionFactory

Essa propriedade é usada para obter o nome do gerenciador de filas ao qual ele está conectado

Quando usado com uma CCDT (Client Channel Definition Table), esse nome pode ser diferente do nome do gerenciador de filas especificado na Connection Factory.

### ***XMSC\_WMQ\_RESOLVED\_QUEUE\_MANAGER\_ID***

**Tipo de dado:**

Sequência

**Propriedade de:**

ConnectionFactory

Essa propriedade é preenchida com o ID do Gerenciador de Filas após a conexão.

### ***XMSC\_WMQ\_SECURITY\_EXIT***

**Tipo de dado:**

Sequência

**Propriedade de:**

ConnectionFactory

Identifica uma saída de segurança do canal.

O valor da propriedade é uma sequência que identifica uma saída de segurança do canal e possui o seguinte formato:

**libraryName**(entryPointNome)

em que,

- **libraryName** é o caminho completo da saída gerenciada .dll
- **entryPointName** é o nome da classe qualificado pelo namespace.

Por exemplo, C:\MySecurityExit.dll(MySecurityExitNameSpace.MySecurityExitClassName)

O comprimento máximo da cadeia é de 128 caracteres.

Por padrão, a propriedade não é configurada

Essa propriedade é relevante somente quando um aplicativo se conecta a um gerenciador de filas no modo de cliente gerenciado Além disso, apenas as saídas gerenciadas são suportadas

### ***XMSC\_WMQ\_SECURITY\_EXIT\_INIT***

**Tipo de dado:**

Sequência

**Propriedade de:**

ConnectionFactory

Os dados do usuário que são transmitidos para uma saída de segurança do canal quando ela é chamada.

O comprimento máximo da sequência de dados do usuário é de 32 caracteres.

Por padrão, a propriedade não é configurada

Essa propriedade é relevante apenas quando um aplicativo se conecta a um gerenciador de filas no modo de cliente gerenciado e a propriedade [“XMSC\\_WMQ\\_SECURITY\\_EXIT”](#) na página 2150 é configurada

### ***XMSC\_WMQ\_SEND\_EXIT***

**Tipo de dado:**

Sequência

**Propriedade de:**

ConnectionFactory

Identifica uma saída de envio de canal.

O valor da propriedade é uma sequência. Uma saída de envio de canal tem o seguinte formato:

**libraryName**(entryPointNome)

em que,

- **libraryName** é o caminho completo da saída gerenciada .dll
- **entryPointName** é o nome da classe qualificado pelo namespace.

Por exemplo, C:\MySendExit.dll(MySendExitNameSpace.MySendExitClassName)

Por padrão, a propriedade não é configurada

Essa propriedade é relevante somente quando um aplicativo se conecta a um gerenciador de filas no modo de cliente gerenciado Além disso, apenas as saídas gerenciadas são suportadas

### ***XMSC\_WMQ\_SEND\_EXIT\_INIT***

**Tipo de dado:**

Sequência

**Propriedade de:**

ConnectionFactory

Os dados do usuário que são transmitidos para as saídas de envio do canal quando são chamadas.

O valor da propriedade é uma sequência de um ou mais itens de dados do usuário separados por vírgula. Por padrão, a propriedade não é configurada

As regras para especificar dados do usuário que são transmitidos para uma sequência de saídas de envio de canal são as mesmas que as regras para especificar dados do usuário que são transmitidos para uma sequência de saídas de recebimento de canal.. Portanto, para obter as regras, consulte [“XMSC\\_WMQ\\_RECEIVE\\_EXIT\\_INIT”](#) na página 2149

Essa propriedade é relevante apenas quando um aplicativo se conecta a um gerenciador de filas no modo de cliente gerenciado e a propriedade [“XMSC\\_WMQ\\_SEND\\_EXIT”](#) na página 2151 é configurada

### ***XMSC\_WMQ\_SEND\_CHECK\_COUNT***

**Tipo de dado:**

System.Int32

**Propriedade de:**

ConnectionFactory

O número de chamadas de envio a serem permitidas entre a verificação de erros de postagem assíncrona, dentro de uma única sessão XMS não transacionada.

Por padrão, esta propriedade é configurada como 0

### ***XMSC\_WMQ\_SHARE\_CONV\_ALLOWED***

**Tipo de dado:**

System.Int32

**Propriedade de:**

ConnectionFactory

**Objetos aplicáveis:**

Nome longo da ferramenta de administração JMS: SHARECONVALLOWED

Nome abreviado da ferramenta de administração JMS: SCALD

Se uma conexão do cliente pode compartilhar seu soquete com outras conexões do XMS de nível superior do mesmo processo para o mesmo gerenciador de filas, se as definições de canal corresponderem. Essa propriedade é fornecida para permitir o isolamento completo de Conexões em soquetes separados, se necessário para desenvolvimento de aplicativos, manutenção ou razões operacionais. A configuração dessa propriedade apenas indica para XMS para tornar o soquete subjacente compartilhado. Ele não indica quantas conexões compartilham um único soquete. O número de conexões que compartilham um soquete é determinado pelo valor SHARECNV que é negociado entre o servidor IBM MQ client e IBM MQ .

Um aplicativo pode configurar as seguintes constantes nomeadas para configurar a propriedade:

- XMSC\_WMQ\_SHARE\_CONV\_ALLOWED\_FALSE-As conexões não compartilham um soquete
- XMSC\_WMQ\_SHARE\_CONV\_ALLOWED\_TRUE-As conexões compartilham um soquete

Por padrão, a propriedade é configurada como XMSC\_WMQ\_SHARE\_CONV\_ALLOWED\_ENABLED

Essa propriedade é relevante somente quando um aplicativo se conecta a um gerenciador de fila no modo de cliente

## ***XMSC\_WMQ\_SSL\_CERT\_STORES***

**Tipo de dado:**

Sequência

**Propriedade de:**

ConnectionFactory

Os locais dos servidores que retêm as listas de revogação de certificado (CRLs) a serem usadas em uma conexão SSL com um gerenciador de filas.

O valor da propriedade é uma lista de uma ou mais URLs separadas por vírgulas. Cada URL possui o seguinte formato:

```
[user[/password]@]ldap://[serveraddress][:portnum][, ...]
```

Esse formato é compatível com, mas estendido do, formato MQJMS básico.

É válido ter um `serveraddress` vazio. Neste caso, XMS assume que o valor é a sequência "localhost".

Uma lista de exemplo é:

```
myuser/mypassword@ldap://server1.mycom.com:389  
ldap://server1.mycom.com  
ldap://  
ldap://:389
```

Para .NET apenas: de IBM MQ 8.0, as conexões gerenciadas para IBM MQ (WMQ\_CM\_CLIENT) e conexões não gerenciadas para IBM MQ (WMQ\_CM\_CLIENT\_UNMANAGED) suportam conexões TLS/SSL.

Por padrão, a propriedade não é configurada

**Conceitos relacionados**

[Suporte ao SSL e TLS para o cliente .NET não gerenciado](#)

[Suporte para SSL e TLS para o cliente .NET gerenciado](#)

## ***XMSC\_WMQ\_SSL\_CIPHER\_SPEC***

### **Tipo de dado:**

Sequência

### **Propriedade de:**

ConnectionFactory

O nome da CipherSpec a ser usada em uma conexão segura com um gerenciador de filas.

As especificações de código que podem ser usadas com o suporte TLS do IBM MQ são listadas na tabela a seguir. Ao exigir um certificado pessoal, você especifica um tamanho de chave para o par de chaves público e particular. O tamanho de chave que é usado durante o handshake SSL é o tamanho armazenado no certificado, a menos que ele seja determinado pelo CipherSpec, conforme indicado na tabela. Por padrão, essa propriedade não está configurada.

<b>Nome do CipherSpec</b>	<b>Protocolo utilizado</b>	<b>Algoritmo hash</b>	<b>Algoritmo de criptografia</b>	<b>Bits de Criptografia</b>	<b>FIPS<sup>1</sup></b>	<b>Conjunto B de 128 bits</b>	<b>Conjunto B de 192 bits</b>
TLS_RSA_WITH_AES_128_CBC_SHA	TLS 1.0	SHA-1	AES	128	Sim	No	No
TLS_RSA_WITH_AES_256_CBC_SHA <sup>2</sup>	TLS 1.0	SHA-1	AES	256	Sim	No	No
TLS_RSA_WITH_DES_CBC_SHA	TLS 1.0	SHA-1	DES	56	No	No	No
TLS_RSA_WITH_3DES_EDE_CBC_SHA <sup>4</sup>	TLS 1.0	SHA-1	3DES	168	Sim	No	No
TLS_RSA_WITH_AES_128_GCM_SHA256	TLS 1.2	SHA-256	AES	128	Sim	No	No
TLS_RSA_WITH_AES_256_GCM_SHA384	TLS 1.2	SHA-384	AES	256	Sim	No	No
TLS_RSA_WITH_AES_128_CBC_SHA256	TLS 1.2	SHA-256	AES	128	Sim	No	No
TLS_RSA_WITH_AES_256_CBC_SHA256	TLS 1.2	SHA-256	AES	256	Sim	No	No
ECDHE_ECDSA_RC4_128_SHA256	TLS 1.2	SHA-256	RC4	128	No	No	No
ECDHE_ECDSA_3DES_EDE_CBC_SHA256	TLS 1.2	SHA-256	3DES	168	Sim	No	No
ECDHE_RSA_RC4_128_SHA256	TLS 1.2	SHA-256	RC4	128	No	No	No
ECDHE_RSA_3DES_EDE_CBC_SHA256	TLS 1.2	SHA-256	3DES	168	Sim	No	No
ECDHE_ECDSA_AES_128_CBC_SHA256	TLS 1.2	SHA-256	AES	128	Sim	No	No
ECDHE_ECDSA_AES_256_CBC_SHA384	TLS 1.2	SHA-384	AES	256	Sim	No	No
ECDHE_RSA_AES_128_CBC_SHA256	TLS 1.2	SHA-256	AES	128	Sim	No	No
ECDHE_RSA_AES_256_CBC_SHA384	TLS 1.2	SHA-384	AES	256	Sim	No	No

Nome do CipherSpec	Protocolo utilizado	Algoritmo hash	Algoritmo de criptografia	Bits de Criptografia	FIPS <sup>1</sup>	Conjunto B de 128 bits	Conjunto B de 192 bits
ECDHE_ECDSA_AES_128_GCM_SHA256	TLS 1.2	SHA-256	AES	128	Sim	Sim	No
ECDHE_ECDSA_AES_256_GCM_SHA384	TLS 1.2	SHA-384	AES	256	Sim	No	Sim
ECDHE_RSA_AES_128_GCM_SHA256	TLS 1.2	SHA-256	AES	128	Sim	No	No
ECDHE_RSA_AES_256_GCM_SHA384	TLS 1.2	SHA-384	AES	256	Sim	No	No
TLS_RSA_WITH_NULL_SHA256	TLS 1.2	SHA-256	Nenhum	0	No	No	No
ECDHE_RSA_NULL_SHA256	TLS 1.2	SHA-256	Nenhum	0	No	No	No
ECDHE_ECDSA_NULL_SHA256	TLS 1.2	SHA-256	Nenhum	0	No	No	No
TLS_RSA_WITH_NULL_NULL	TLS 1.2	Nenhum	Nenhum	0	No	No	No
TLS_RSA_WITH_RC4_128_SHA256	TLS 1.2	SHA-256	RC4	128	No	No	No

**Notas:**

1. Especifica se o CipherSpec está em conformidade com o Federal Information Processing Standards (FIPS) 140-2. Para obter uma explicação do FIPS e informações sobre como configurar o IBM MQ para operação compatível com FIPS 140-2, consulte [Federal Information Processing Standards \(FIPS\)](#).
2. Esse CipherSpec não pode ser usado para proteger uma conexão do IBM MQ Explorer com um gerenciador de filas, a menos que os arquivos de políticas irrestritos apropriados sejam aplicados no JRE usado pelo IBM MQ Explorer
3. Este CipherSpec era certificado FIPS 140-2 antes de 19 de Maio de 2007.
4. Quando o IBM MQ é configurado para operação em conformidade com FIPS 140-2, este CipherSpec pode ser usado para transferir até 32 GB de dados antes da conexão ser finalizada com o erro AMQ9288. Para evitar esse erro, evite usar DES triplo (que foi descontinuado) ou ative a reconfiguração de chave secreta ao usar esse CipherSpec em uma configuração FIPS 140-2.

**Conceitos relacionados**

[Integridade de dados de mensagens](#)

**Tarefas relacionadas**

[Assegurando](#)

[Especificando CipherSpecs](#)

***XMSC\_WMQ\_SSL\_CIPHER\_SUITE***

**Tipo de dado:**

Sequência

**Propriedade de:**

ConnectionFactory

O nome do CipherSuite a ser usado em uma conexão TLS para um gerenciador de filas. O protocolo usado para negociar a conexão segura depende do CipherSuite especificado.

Essa propriedade possui os seguintes valores canônicos:

- SSL\_RSA\_WITH\_DES\_CBC\_SHA
- SSL\_RSA\_EXPORT1024\_WITH\_DES\_CBC\_SHA
- SSL\_RSA\_EXPORT1024\_WITH\_RC4\_56\_SHA
- SSL\_RSA\_EXPORT\_WITH\_RC4\_40\_MD5
- SSL\_RSA\_WITH\_RC4\_128\_MD5
- SSL\_RSA\_WITH\_RC4\_128\_SHA
- SSL\_RSA\_WITH\_3DES\_EDE\_CBC\_SHA
- SSL\_RSA\_WITH\_AES\_128\_CBC\_SHA
- SSL\_RSA\_WITH\_AES\_256\_CBC\_SHA
- SSL\_RSA\_WITH\_DES\_CBC\_SHA
- SSL\_RSA\_WITH\_3DES\_EDE\_CBC\_SHA

Esse valor pode ser fornecido como uma alternativa para `XMSC_WMQ_SSL_CIPHER_SPEC`.

Se um valor não vazio for especificado para `XMSC_WMQ_SSL_CIPHER_SPEC`, esse valor substituirá a configuração para `XMSC_WMQ_SSL_CIPHER_SUITE`. Se `XMSC_WMQ_SSL_CIPHER_SPEC` não tiver um valor, o valor de `XMSC_WMQ_SSL_CIPHER_SUITE` será usado como o conjunto de cifras a ser fornecido para IBM Global Security Kit (GSKit). Nesse caso, o valor é mapeado para o valor CipherSpec equivalente, conforme descrito em mapeamentos de nome CipherSuite e CipherSpec para conexões do XMS para um IBM MQ gerenciador de filas.

Se `XMSC_WMQ_SSL_CIPHER_SPEC` e `XMSC_WMQ_SSL_CIPHER_SUITE` estiverem vazios, o campo `pChDef->SSLCipherSpec` será preenchido com espaços.

Para .NET apenas: de IBM MQ 8.0, as conexões gerenciadas para IBM MQ (`WMQ_CM_CLIENT`) e conexões não gerenciadas para IBM MQ (`WMQ_CM_CLIENT_UNMANAGED`) suportam conexões TLS/SSL.

Por padrão, a propriedade não é configurada

### **Conceitos relacionados**

Suporte ao SSL e TLS para o cliente .NET não gerenciado

Suporte para SSL e TLS para o cliente .NET gerenciado

## ***XMSC\_WMQ\_SSL\_CRYPTO\_HW***

### **Tipo de dado:**

Sequência

### **Propriedade de:**

ConnectionFactory

Detalhes de configuração para o hardware criptográfico conectado ao sistema do cliente.

Essa propriedade possui os seguintes valores canônicos:

- GSK\_ACCELERATOR\_RAINBOW\_CS\_OFF
- GSK\_ACCELERATOR\_RAINBOW\_CS\_ON
- GSK\_ACCELERATOR\_NCIPHER\_NF\_OFF
- GSK\_ACCELERATOR\_NCIPHER\_NF\_ON

Há um formato especial para o hardware criptográfico PKCS11 (em que `DriverPath`, `TokenLabel` e `TokenPassword` são sequências especificadas pelo usuário):

```
GSK_PKCS11=PKCS#11 DriverPath; PKCS#11 TokenLabel;PKCS#11 TokenPassword
```

O XMS não interpreta ou altera o conteúdo da sequência. Ele copia o valor fornecido, até um limite de 256 caracteres de byte único, no `MQSCO MQSCO.CryptoHardware` de `CryptoHardware`

Para .NET apenas: de IBM MQ 8.0, as conexões gerenciadas para IBM MQ (WMQ\_CM\_CLIENT) e conexões não gerenciadas para IBM MQ (WMQ\_CM\_CLIENT\_UNMANAGED) suportam conexões TLS/SSL.

Por padrão, a propriedade não é configurada

#### **Conceitos relacionados**

[Suporte ao SSL e TLS para o cliente .NET não gerenciado](#)

[Suporte para SSL e TLS para o cliente .NET gerenciado](#)

### ***XMSC\_WMQ\_SSL\_FIPS\_REQUIRED***

#### **Tipo de dado:**

Booleana

#### **Propriedade de:**

ConnectionFactory

O valor dessa propriedade determina se um aplicativo pode ou não usar conjuntos de cifras compatíveis não FIPS. Se essa propriedade for configurada como true, apenas algoritmos do FIPS serão usados para a conexão cliente-servidor.

Essa propriedade pode ter os valores a seguir, que são convertidos para os dois valores canônicos para o MQSCO MQSCO.FipsRequired:

<i>Tabela 880. Tabela de valores para MQSCO.FlipsRequired</i>		
<b>Value</b>	<b>Descrição</b>	<b>Valor correspondente de MQSCO.FipsRequired</b>
false	Qualquer CipherSpec pode ser usado	MQSSL_FIPS_NO (o padrão),
true	Somente algoritmos criptográficos certificados pelo FIPS podem ser usados no CipherSpec que se aplica a essa conexão do cliente	MQSSL_FIPS_YES

XMS copia o valor relevante no MQSCO.FipsRequired antes de chamar MQCONN.

Para .NET apenas: de IBM MQ 8.0, as conexões gerenciadas para IBM MQ (WMQ\_CM\_CLIENT) e conexões não gerenciadas para IBM MQ (WMQ\_CM\_CLIENT\_UNMANAGED) suportam conexões TLS/SSL.

#### **Conceitos relacionados**

[Suporte ao SSL e TLS para o cliente .NET não gerenciado](#)

[Suporte para SSL e TLS para o cliente .NET gerenciado](#)

### ***XMSC\_WMQ\_SSL\_KEY\_REPOSITORY***

#### **Tipo de dado:**

Sequência

#### **Propriedade de:**

ConnectionFactory

O local do arquivo do banco de dados de chaves no qual chaves e certificados são armazenados.

XMS copia a sequência, até um limite de 256 caracteres de byte único, no MQSCO.KeyRepository do KeyRepository IBM MQ interpreta essa sequência como um nome de arquivo, incluindo o caminho completo..

Para .NET apenas: de IBM MQ 8.0, as conexões gerenciadas para IBM MQ (WMQ\_CM\_CLIENT) e conexões não gerenciadas para IBM MQ (WMQ\_CM\_CLIENT\_UNMANAGED) suportam conexões TLS/SSL.

Por padrão, a propriedade não é configurada

#### **Conceitos relacionados**

[Suporte ao SSL e TLS para o cliente .NET não gerenciado](#)



## ***XMSC\_WMQ\_SSL\_KEY\_RESETCOUNT***

**Tipo de dado:**

System.Int32

**Propriedade de:**

ConnectionFactory

O KeyResetCount representa o número total de bytes não criptografados enviados e recebidos dentro de uma conversa SSL antes de a chave secreta ser renegociada. O número de bytes inclui informações de controle enviadas pelo MCA.

XMS copia o valor fornecido para essa propriedade no MQSCO.KeyResetCount antes de chamar MQCONN.

O parâmetro MQSCO.KeyRestCount está disponível apenas a partir do IBM MQ versão 6 Se IBM MQ versão 5.3, se essa propriedade for configurada, o XMS não tentará fazer a conexão com o gerenciador de fila e, em vez disso, emitirá uma exceção apropriada

Para .NET apenas: de IBM MQ 8.0, as conexões gerenciadas para IBM MQ (WMQ\_CM\_CLIENT) e conexões não gerenciadas para IBM MQ (WMQ\_CM\_CLIENT\_UNMANAGED) suportam conexões TLS/SSL.

O valor padrão dessa propriedade é zero, o que significa que chaves secretas nunca são renegociadas.

**Conceitos relacionados**

[Suporte ao SSL e TLS para o cliente .NET não gerenciado](#)

[Suporte para SSL e TLS para o cliente .NET gerenciado](#)

## ***XMSC\_WMQ\_SSL\_PEER\_NAME***

**Tipo de dado:**

Sequência

**Propriedade de:**

ConnectionFactory

O nome do peer a ser usado em uma conexão SSL com um gerenciador de filas.

Não há nenhuma lista de valores canônicos para esta propriedade Em vez disso, deve-se construir essa cadeia de acordo com as regras para SSLPEER

Um exemplo de nome de peer é:

```
"CN=John Smith, O=IBM ,OU=Test , C=GB"
```

XMS copia a sequência na página de códigos de byte único correta e coloca os valores corretos em MQCD.SSLPeerNamePtr e MQCD.SSLPeerNameLength antes de chamar MQCONN.

Essa propriedade será relevante apenas se o aplicativo se conectar a um gerenciador de filas em modo cliente

Para .NET apenas: de IBM MQ 8.0, as conexões gerenciadas para IBM MQ (WMQ\_CM\_CLIENT) e conexões não gerenciadas para IBM MQ (WMQ\_CM\_CLIENT\_UNMANAGED) suportam conexões TLS/SSL.

Por padrão, a propriedade não é configurada

**Conceitos relacionados**

[Suporte ao SSL e TLS para o cliente .NET não gerenciado](#)

[Suporte para SSL e TLS para o cliente .NET gerenciado](#)

**Referências relacionadas**

[SSLPEERNAME](#)

## ***XMSC\_WMQ\_SYNCPOINT\_ALL\_GETS***

**Tipo de dado:**

System.Boolean

**Propriedade de:**

ConnectionFactory

Se todas as mensagens devem ser recuperadas de filas dentro do controle de ponto de sincronização.

Os valores válidos da propriedade são os seguintes:

<b>Valor válido</b>	<b>Significado</b>
false	Quando as circunstâncias forem apropriadas, o cliente XMS poderá recuperar mensagens de filas fora do controle do ponto de sincronização..
true	O cliente XMS deve recuperar todas as mensagens das filas dentro do controle de ponto de sincronização.

O valor padrão é falso.

## ***XMSC\_WMQ\_TARGET\_CLIENT***

**Tipo de dado:**

System.Int32

**Propriedade de:**

Destino

**Nome usado em um URI:**

targetClient

Se as mensagens enviadas para o destino contêm um cabeçalho MQRFH2.

Se um aplicativo enviar uma mensagem contendo um cabeçalho MQRFH2 , o aplicativo de recepção deverá ser capaz de manipular o cabeçalho.

Os valores válidos da propriedade são os seguintes:

<b>Valor válido</b>	<b>Significado</b>
XMSC_WMQ_TARGET_DEST_JMS	As mensagens enviadas ao destino contêm um cabeçalho MQRFH2 . Especifique esse valor se o aplicativo estiver enviando as mensagens para outro aplicativo XMS , um aplicativo IBM MQ classes for JMS ou um aplicativo IBM MQ nativo projetado para manipular um cabeçalho MQRFH2 .
XMSC_WMQ_TARGET_DEST_MQ	As mensagens enviadas ao destino não contêm um cabeçalho MQRFH2 . Especifique esse valor se o aplicativo estiver enviando as mensagens para um aplicativo IBM MQ nativo que não foi projetado para manipular um cabeçalho MQRFH2 .

O valor padrão é XMSC\_WMQ\_TARGET\_DEST\_JMS.

## ***XMSC\_WMQ\_TEMP\_Q\_PREFIX***

**Tipo de dado:**

Sequência

**Propriedade de:**

ConnectionFactory

O prefixo usado para formar o nome da fila dinâmica IBM MQ que é criada quando o aplicativo cria uma fila temporária XMS .

As regras para formar o prefixo são as mesmas que as regras para formar o conteúdo do campo **DynamicQName** em um descritor de objeto, mas o último caractere não em branco deve ser um asterisco (\*). Se a propriedade não for configurada, o valor usado será CSQ.\* on z/OS e AMQ.\* nas outras plataformas.. Por padrão, a propriedade não é configurada

Essa propriedade é relevante apenas no domínio de ponto a ponto

### ***XMSC\_WMQ\_TEMP\_TOPIC\_PREFIX***

**Tipo de dado:**

Sequência

**Propriedade de:**

ConnectionFactory, Destino

Ao criar tópicos temporários, o XMS gera uma sequência de tópicos no formato "TEMP/TEMPTOPICPREFIX/unique\_id" ou se essa propriedade contiver o valor padrão, essa sequência, "TEMP/unique\_id", será gerada. Especificar um valor não vazio permite que as filas modelo específicas sejam definidas para criar as filas gerenciadas para assinantes de tópicos temporários criados sob essa conexão.

Qualquer sequência não nula que consiste apenas em caracteres válidos para uma sequência de tópicos IBM MQ é um valor válido para essa propriedade.

Por padrão, essa propriedade é configurada como "" (sequência vazia).

**Nota:** Essa propriedade é relevante apenas no domínio de publicação / assinatura

### ***XMSC\_WMQ\_TEMPORARY\_MODEL***

**Tipo de dado:**

Sequência

**Propriedade de:**

ConnectionFactory

O nome da fila modelo IBM MQ a partir da qual uma fila dinâmica é criada quando o aplicativo cria uma XMS fila temporária.

O valor padrão da propriedade é SYSTEM.DEFAULT.MODEL.QUEUE.

Essa propriedade é relevante apenas no domínio de ponto a ponto

### ***XMSC\_WMQ\_WILDCARD\_FORMAT***

**Tipo de dado:**

System.Int32

**Propriedade de:**

ConnectionFactory, Destino

Essa propriedade determina qual versão de sintaxe curinga deve ser usada.

Ao usar a publicação / assinatura com IBM MQ '\*' e '?' são tratados como curingas Considerando que '#' e '+' são tratados como curingas ao usar publicar assinatura com IBM Integration Bus. Essa propriedade substitui a propriedade XMSC\_WMQ\_BROKER\_VERSION..

Os valores válidos para essa propriedade são:

#### **XMSC\_WMQ\_WILDCARD\_TOPIC\_ONLY**

Reconhece os curingas de nível de tópico apenas, ou seja, '#' e '+' são tratados como curingas Esse valor é igual a XMSC\_WMQ\_BROKER\_V2.

#### **XMSC\_WMQ\_WILDCARD\_CHAR\_ONLY**

Reconhece os caracteres curingas apenas, ou seja, '\*' e '?' são tratados como curingas Esse valor é igual a XMSC\_WMQ\_BROKER\_V1.

Por padrão, essa propriedade é configurada para XMSC\_WMQ\_WILDCARD\_TOPIC\_ONLY

## ***XMSC\_WPM\_BUS\_NAME***

### **Tipo de dado:**

Sequência

### **Propriedade de:**

ConnectionFactory e Destino

### **Nome usado em um URI:**

busName

Para um connection factory, o nome do barramento de integração de serviços ao qual o aplicativo se conecta ou, para um destino, o nome do barramento de integração de serviços no qual o destino existe.

Para um destino que é um tópico, essa propriedade é o nome do barramento de integração de serviços no qual o espaço de tópicos associado existe... Este espaço de tópico é especificado pelo [XMSC\\_WPM\\_TOPIC\\_SPACE](#) de propriedade

Se a propriedade não for configurada para um destino, a fila ou o espaço de tópico associado será considerado existente no barramento de integração de serviços ao qual o aplicativo se conecta.

Por padrão, a propriedade não é configurada

## ***XMSC\_WPM\_CONNECTION\_PROTOCOL***

### **Tipo de dado:**

System.Int32

### **Propriedade de:**

Conexão

O protocolo de comunicações usado para a conexão com o mecanismo do sistema de mensagens. Esta propriedade é somente leitura.

Os valores possíveis da propriedade são os seguintes:

<b>Value</b>	<b>Significado</b>
XMSC_WPM_CP_HTTP	A conexão usa HTTP sobre TCP/IP..
XMSC_WPM_CP_TCP	A conexão usa TCP/IP.

## ***XMSC\_WPM\_CONNECTION\_PROXIMIDADE***

### **Tipo de dado:**

System.Int32

### **Propriedade de:**

ConnectionFactory

A configuração de proximidade de conexão para a conexão. Essa propriedade determina o quão próximo o mecanismo do sistema de mensagens ao qual o aplicativo se conecta deve estar do servidor de autoinicialização..

Os valores válidos da propriedade são os seguintes:

<b>Valor válido</b>	<b>Configuração de proximidade de conexão..</b>
XMSC_WPM_CONNECTION_PROXIMITY_BUS	Barramento
XMSC_WPM_CONNECTION_PROXIMITY_CLUSTER	Cluster
XMSC_WPM_CONNECTION_PROXIMITY_HOST	Host
XMSC_WPM_CONNECTION_PROXIMITY_SERVER	Servidor

O valor padrão é XMSC\_WPM\_CONNECTION\_PROXIMITY\_BUS.

## ***XMSC\_WPM\_DUR\_SUB\_HOME***

**Tipo de dado:**

Sequência

**Propriedade de:**

ConnectionFactory

**Nome usado em um URI:**

durableSubscriptionInício

O nome do mecanismo do sistema de mensagens no qual todas as assinaturas duráveis para uma conexão ou um destino são gerenciadas. Mensagens a serem entregues aos assinantes duráveis são armazenadas no ponto de publicação do mesmo mecanismo do sistema de mensagens.

Um lar de assinaturas duráveis deve ser especificado para uma conexão antes que um aplicativo possa criar um assinante durável que use a conexão. Qualquer valor especificado para um destino substitui o valor especificado para a conexão.

Por padrão, a propriedade não é configurada

Essa propriedade é relevante apenas no domínio de publicação / assinatura

## ***XMSC\_WPM\_HOST\_NAME***

**Tipo de dado:**

Sequência

**Propriedade de:**

Conexão

O nome do host ou o endereço IP do sistema que contém o mecanismo do sistema de mensagens para o qual o aplicativo está conectado. Esta propriedade é somente leitura.

## ***XMSC\_WPM\_LOCAL\_ADDRESS***

**Tipo de dado:**

Sequência

**Propriedade de:**

ConnectionFactory

Para uma conexão com um barramento de integração de serviços, essa propriedade especifica a interface de rede local a ser usada, a porta local ou o intervalo de portas locais a serem usados ou ambos.

O valor da propriedade é uma sequência com o seguinte formato:

*[host\_name] [(low\_port) [,high\_port]]*

Os significados das variáveis são os seguintes:

***host\_name***

O nome do host ou o endereço IP da interface de rede local a ser usado para a conexão

Fornecer essas informações é necessário apenas se o sistema no qual o aplicativo está em execução tiver duas ou mais interfaces de rede e você precisar ser capaz de especificar qual interface deve ser usada para a conexão.. Se o sistema tiver apenas uma interface de rede, somente essa interface poderá ser usada Se o sistema tiver duas ou mais interfaces de rede e você não especificar qual interface deve ser usada, a interface será selecionada aleatoriamente

***low\_port***

O número da porta local a ser usada para a conexão

Se *high\_port* também for especificado, *low\_port* será interpretado como o número da porta mais baixo em um intervalo de números de portas

### **high\_port**

O número da porta mais alto em um intervalo de números de porta. Uma das portas no intervalo especificado deve ser usada para a conexão..

Aqui estão alguns exemplos de valores válidos da propriedade:

JÚPITER  
9.20.4.98  
JUPITER (1000)  
9.20.4.98(1000,2000)  
(1000)  
(1000,2000)

Por padrão, a propriedade não é configurada

### ***XMSC\_WPM\_ME\_NAME***

#### **Tipo de dado:**

Sequência

#### **Propriedade de:**

Conexão

O nome do mecanismo do sistema de mensagens para o qual o aplicativo está conectado. Esta propriedade é somente leitura.

### ***XMSC\_WPM\_NON\_PERSISTENT\_MAP***

#### **Tipo de dado:**

System.Int32

#### **Propriedade de:**

ConnectionFactory

O nível de confiabilidade de mensagens não persistentes que são enviadas usando a conexão.

Os valores válidos da propriedade são os seguintes:

#### **Valor válido**

XMSC\_WPM\_MAPPING\_AS\_DESTINATION

XMSC\_WPM\_MAPPING\_BEST\_SEM esforço  
PERSISTENT

XMSC\_WPM\_MAPPING\_EXPRESS\_NON\_  
PERSISTENT

XMSC\_WPM\_MAPPING\_RELIABLE\_NON\_  
PERSISTENT

XMSC\_WPM\_MAPPING\_RELIABLE\_PERSISTENT

XMSC\_WPM\_MAPPING\_ASSURED\_PERSISTENT

#### **Nível de confiabilidade**

Determinado pelo nível de confiabilidade padrão especificado para a fila ou espaço de tópicos no barramento de integração de serviços

Best effort nonpersistent

Express nonpersistent

Confiável Não Persistente

Persistente confiável

Garantido Persistente

O valor padrão é XMSC\_WPM\_MAPPING\_EXPRESS\_NON\_PERSISTENT.

## ***XMSC\_WPM\_PERSISTENT\_MAP***

**Tipo de dado:**

System.Int32

**Propriedade de:**

ConnectionFactory

O nível de confiabilidade de mensagens persistentes que são enviadas usando a conexão.

Os valores válidos da propriedade são os seguintes:

**Valor válido**

XMSC\_WPM\_MAPPING\_AS\_DESTINATION

XMSC\_WPM\_MAPPING\_BEST\_SEM esforço  
PERSISTENT

XMSC\_WPM\_MAPPING\_EXPRESS\_NON\_  
PERSISTENT

XMSC\_WPM\_MAPPING\_RELIABLE\_NON\_  
PERSISTENT

XMSC\_WPM\_MAPPING\_RELIABLE\_PERSISTENT

XMSC\_WPM\_MAPPING\_ASSURED\_PERSISTENT

**Nível de confiabilidade**

Determinado pelo nível de confiabilidade padrão especificado para a fila ou espaço de tópicos no barramento de integração de serviços

Best effort nonpersistent

Express nonpersistent

Confiável Não Persistente

Persistente confiável

Garantido Persistente

O valor padrão é XMSC\_WPM\_MAPPING\_RELIABLE\_PERSISTENT.

## ***XMSC\_WPM\_PORT***

**Tipo de dado:**

System.Int32

**Propriedade de:**

Conexão

O número da porta atendida pelo mecanismo do sistema de mensagens para o qual o aplicativo está conectado. Esta propriedade é somente leitura.

## ***XMSC\_WPM\_PROVIDER\_ENDPOINTS***

**Tipo de dado:**

Sequência

**Propriedade de:**

ConnectionFactory

Uma sequência de um ou mais endereços de terminal de servidores de autoinicialização. Os endereços de terminal são separados por vírgula.

Um servidor de autoinicialização é um servidor de aplicativos responsável por selecionar o mecanismo do sistema de mensagens ao qual o aplicativo se conecta. O endereço do terminal de um servidor de autoinicialização tem o seguinte formato:

*host\_name:port\_number:chain\_name*

Os significados dos componentes de um endereço de terminal são os seguintes:

**host\_name**

O nome do host ou endereço IP do sistema no qual o servidor de autoinicialização reside. Se nenhum nome do sistema central ou endereço IP for especificado, o padrão será localhost.

**port\_number**

O número da porta na qual o servidor de autoinicialização atende solicitações recebidas. Se nenhum número de porta for especificado, o padrão será 7276..

**chain\_name**

O nome de uma cadeia de transporte de autoinicialização usada pelo servidor de autoinicialização. Os valores válidos são os seguintes:

Valor válido	Nome da cadeia de transporte de autoinicialização
XMSC_WPM_BOOTSTRAP_HTTP	Sistema de Mensagens BootstrapTunneled
XMSC_WPM_BOOTSTRAP_HTTPS	BootstrapTunneledSecureMessaging
XMSC_WPM_BOOTSTRAP_SSL	Sistema de Mensagens BootstrapSecure
XMSC_WPM_BOOTSTRAP_TCP	Sistema de Mensagens BootstrapBasic

Se nenhum nome for especificado, o valor padrão será XMSC\_WPM\_BOOTSTRAP\_TCP.

Se nenhum endereço de terminal for especificado, o padrão será localhost:7276:BootstrapBasicMessaging.

**XMSC\_WPM\_SSL\_CIPHER\_SUITE****Tipo de dado:**

Sequência de caracteres

**Propriedade de:**

ConnectionFactory

O nome do CipherSuite a ser usado em uma conexão TLS com um mecanismo do sistema de mensagens do WebSphere Application Server service integration bus O protocolo usado para negociar a conexão segura depende do CipherSuite especificado.

<i>Tabela 881. Opções CipherSuite para conexão com um mecanismo do sistema de mensagens do WebSphere Application Server service integration bus</i>	
Conjunto de cifras	Protocolo utilizado
TLS_RSA_WITH_DES_CBC_SHA	TLSv1
TLS_RSA_WITH_3DES_EDE_CBC_SHA	TLSv1
TLS_RSA_WITH_AES_128_CBC_SHA	TLSv1
TLS_RSA_WITH_AES_256_CBC_SHA	TLSv1

**Observações:**

- Windows** TLS\_RSA\_WITH\_AES\_128\_CBC\_SHA e TLS\_RSA\_WITH\_AES\_256\_CBC\_SHA CipherSuites são suportados no Windows apenas. (Isso é ditado por GSKit.)
- Deprecated** TLS\_RSA\_WITH\_3DES\_EDE\_CBC\_SHA foi descontinuado No entanto, ele ainda pode ser usado para transferir até 32 GB de dados antes de a conexão ser finalizada com erro AMQ9288. Para evitar esse erro, você precisará evitar o uso de DES triplo ou ativar a reconfiguração de chave secreta ao usar esse CipherSpec.

Não há nenhum padrão para essa propriedade Se você desejar usar SSL ou TLS, deverá especificar um valor para essa propriedade, caso contrário, seu aplicativo não poderá se conectar com êxito ao servidor.



## ***XMSC\_WPM\_SSL\_FIPS\_REQUIRED***

**Nota:** No AIX, Linux, and Windows, IBM MQ fornece conformidade FIPS 140-2 por meio do módulo criptográfico IBM Crypto for C (ICC) . O certificado deste módulo foi movido para o status Histórico. Os clientes devem visualizar o [IBM Crypto for C \(ICC\) certificado](#) e estar ciente de qualquer aviso fornecido pelo NIST Um módulo FIPS 140-3 de substituição está atualmente em andamento e seu status pode ser visualizado procurando por ele na [NIST CMVP modules in process list](#).

### **Tipo de dado:**

Booleana

### **Propriedade de:**

ConnectionFactory

O valor dessa propriedade determina se um aplicativo pode ou não usar conjuntos de cifras compatíveis não FIPS. Se essa propriedade for configurada como true, apenas algoritmos FIPS serão usados para a conexão cliente / servidor. Configurar o valor dessa propriedade como TRUE impede que o aplicativo use conjuntos de cifras não compatíveis com FIPS.

Por padrão, a propriedade é configurada como FALSE (ou seja, modo FIPS desativado).

## ***XMSC\_WPM\_SSL\_KEY\_REPOSITORY***

### **Tipo de dado:**

Sequência

### **Propriedade de:**

ConnectionFactory

Um caminho para o arquivo que é o arquivo do conjunto de chaves contendo as chaves públicas ou privadas a serem usadas na conexão segura.

A configuração da propriedade do arquivo de conjunto de chaves para o valor especial de XMSC\_WPM\_SSL\_MS\_CERT\_STORE especifica o uso do banco de dados de chaves do Microsoft Windows . Usar o banco de dados de chave do Microsoft Windows , localizado em **Painel de Controle > Opções da Internet > Conteúdo > Certificados**, remove a necessidade de um banco de dados de arquivo de chave separado. O uso dessa constante no Windows x64 e outras plataformas não é permitido.

Por padrão, a propriedade não é configurada

## ***XMSC\_WPM\_SSL\_KEYRING\_LABEL***

### **Tipo de dado:**

Sequência

### **Propriedade de:**

ConnectionFactory

O certificado a ser usado ao autenticar-se com o servidor. Se nenhum valor for especificado, o certificado padrão será usado

Por padrão, a propriedade não é configurada

## ***XMSC\_WPM\_SSL\_KEYRING\_PW***

### **Tipo de dado:**

Sequência

### **Propriedade de:**

ConnectionFactory

A senha do arquivo do conjunto de chaves.

Esta propriedade pode ser usada como uma alternativa para usar [XMSC\\_WPM\\_SSL\\_KEYRING\\_STASH\\_FILE](#) para configurar a senha para o arquivo de conjunto de chaves.

Por padrão, a propriedade não é configurada

## ***XMSC\_WPM\_SSL\_KEYRING\_STASH\_FILE***

**Tipo de dado:**

Sequência

**Propriedade de:**

ConnectionFactory

O nome de um arquivo binário contendo a senha do arquivo do repositório de chaves.

Essa propriedade pode ser usada como uma alternativa para usar [XMSC\\_WPM\\_SSL\\_KEYRING\\_PW](#) para configurar a senha para o arquivo de conjunto de chaves.

Por padrão, a propriedade não é configurada

## ***XMSC\_WPM\_TARGET\_GROUP***

**Tipo de dado:**

Sequência

**Propriedade de:**

ConnectionFactory

O nome de um grupo de destinos de mecanismos do sistema de mensagens. A natureza do grupo de destino é determinada pelo [XMSC\\_WPM\\_TARGET\\_TYPE](#) de propriedade

Configure esta propriedade se desejar restringir a procura de um mecanismo do sistema de mensagens para um subgrupo dos mecanismos de sistema de mensagens no barramento de integração de serviços. Se você desejar que seu aplicativo possa se conectar a qualquer mecanismo do sistema de mensagens no barramento de integração de serviços, não configure essa propriedade.

Por padrão, a propriedade não é configurada

## ***XMSC\_WPM\_TARGET\_SIGNIFICATIVO***

**Tipo de dado:**

System.Int32

**Propriedade de:**

ConnectionFactory

O significado do grupo de destinos dos mecanismos do sistema de mensagens.

Os valores válidos da propriedade são os seguintes:

<b>Valor válido</b>	<b>Significado</b>
XMSC_WPM_TARGET_SIGNIFICANCE_ Preferido	Um mecanismo do sistema de mensagens no grupo de destino será selecionado, se um estiver disponível. Caso contrário, um mecanismo do sistema de mensagens fora do grupo de destino será selecionado, contanto que ele esteja no mesmo barramento de integração de serviços.
XMSC_WPM_TARGET_SIGNIFICANCE_ REQUIRED	O mecanismo do sistema de mensagens selecionado deve estar no grupo de destinos. Se um mecanismo do sistema de mensagens no grupo de destino não estiver disponível, o processo de conexão falhará.

O valor padrão da propriedade é XMSC\_WPM\_TARGET\_SIGNIFICANCE\_PREFERRED.

## ***XMSC\_WPM\_TARGET\_TRANSPORT\_CHAIN***

**Tipo de dado:**

Sequência

**Propriedade de:**

ConnectionFactory

O nome da cadeia de transporte de entrada que o aplicativo deve usar para se conectar a um mecanismo do sistema de mensagens.

O valor da propriedade pode ser o nome de qualquer cadeia de transporte de entrada disponível no servidor de aplicativos que hospeda o mecanismo do sistema de mensagens. A constante nomeada a seguir é fornecida para uma das cadeias de transporte de entrada predefinidas:

<b>Constante nomeada</b>	<b>Nome da cadeia de transporte</b>
XMSC_WPM_TARGET_TRANSPORT_CHAIN_BASIC	Sistema de Mensagens InboundBasic

O valor padrão da propriedade é XMSC\_WPM\_TARGET\_TRANSPORT\_CHAIN\_BASIC.

***XMSC\_WPM\_TARGET\_TYPE*****Tipo de dado:**

System.Int32

**Propriedade de:**

ConnectionFactory

O tipo do grupo de destinos de mecanismos do sistema de mensagens. Essa propriedade determina a natureza do grupo de destino identificado pela propriedade [XMSC\\_WPM\\_TARGET\\_GROUP](#)

Os valores válidos da propriedade são os seguintes:

<b>Valor válido</b>	<b>Significado</b>
XMSC_WPM_TARGET_TYPE_BUSMEMBER	O nome do grupo de destino é o nome de um membro do barramento O grupo de destino é todos os mecanismos do sistema de mensagens no membro do barramento
XMSC_WPM_TARGET_TYPE_CUSTOM	O nome do grupo de destino é o nome de um grupo definido pelo usuário de mecanismos do sistema de mensagens O grupo de destino é todos os mecanismos do sistema de mensagens registrados com o grupo definido pelo usuário.
XMSC_WPM_TARGET_TYPE_ME	O nome do grupo de destino é o nome de um mecanismo do mensagens. O grupo de destino é o mecanismo do sistema de mensagens especificado

Por padrão, a propriedade não é configurada

***XMSC\_WPM\_TEMP\_Q\_PREFIX*****Tipo de dado:**

Sequência

**Propriedade de:**

ConnectionFactory

O prefixo usado para formar o nome da fila temporária criada no barramento de integração de serviços quando o aplicativo cria uma XMS fila temporária. O prefixo pode conter até 12 caracteres.

O nome de uma fila temporária começa com os caracteres "\_Q" seguidos pelo prefixo. O restante do nome consiste em caracteres gerados pelo sistema..

Por padrão, a propriedade não é configurada, o que significa que o nome de uma fila temporária não tem um prefixo

Essa propriedade é relevante apenas no domínio de ponto a ponto

### ***XMSC\_WPM\_TEMP\_TOPIC\_PREFIX***

**Tipo de dado:**

Sequência

**Propriedade de:**

ConnectionFactory

O prefixo usado para formar o nome de um tópico temporário que é criado pelo aplicativo. O prefixo pode conter até 12 caracteres.

O nome de um tópico temporário começa com os caracteres "\_T" seguidos pelo prefixo. O restante do nome consiste em caracteres gerados pelo sistema..

Por padrão, a propriedade não é configurada, o que significa que o nome de um tópico temporário não possui um prefixo..

Essa propriedade é relevante apenas no domínio de publicação / assinatura

### ***XMSC\_WPM\_TOPIC\_SPACE***

**Tipo de dado:**

Sequência

**Propriedade de:**

Destino

**Nome usado em um URI:**

topicSpace

O nome do espaço de tópico que contém o tópico. Somente um destino que é um tópico pode ter essa propriedade

Por padrão, a propriedade não é configurada, o que significa que o espaço de tópico padrão é assumido

Essa propriedade é relevante apenas no domínio de publicação / assinatura

## **Managed File Transfer Desenvolvendo referência de aplicativos**

---

Informações de referência para ajudar a desenvolver aplicativos para o Managed File Transfer.

### **Exemplos de Uso de `fteCreateTransfer` para Iniciar Programas**

É possível usar o comando **`fteCreateTransfer`** para especificar programas para execução antes ou após a transferência.

Além do uso de **`fteCreateTransfer`**, existem outras maneiras de chamar um programa antes ou depois de uma transferência. Para obter mais informações, consulte [Especificando programas a serem executados com o MFT](#)

Todos estes exemplos usam a seguinte sintaxe para especificar um programa:

```
[type:]commandspec[, [retrycount][, [retrywait][, successrc]]]
```

Para obter mais informações sobre essa sintaxe, consulte [\*\*`fteCreateTransfer`\*\*: iniciar uma nova transferência de arquivos](#)

## Executando um Programa Executável

O exemplo a seguir especifica um programa executável chamado `mycommand` e transmite dois argumentos, `a` e `b`, para o programa.

```
mycommand(a,b)
```

Para executar este programa no agente de origem `AGENT1` antes do início da transferência, use o seguinte comando:

```
fteCreateTransfer -sa AGENT1 -da AGENT2 -presrc mycommand(a,b)
destinationSpecification sourceSpecification
```

## Executando e Tentado Novamente um Programa Executável

O exemplo a seguir especifica um programa executável chamado `simple`, que não usa nenhum argumento. Um valor de 1 é especificado para `retrycount` e um valor de 5 é especificado para `retrywait`. Estes valores significam que o programa será tentado novamente uma vez, se não retornar um código de retorno bem-sucedido, após uma espera de cinco segundos. Nenhum valor foi especificado para `successrc`, portanto, o único código de retorno bem-sucedido é o valor padrão de 0.

```
executable:simple,1,5
```

Para executar este programa no agente de origem `AGENT1` após a conclusão da transferência, use o seguinte comando:

```
fteCreateTransfer -sa AGENT1 -da AGENT2 -postsrc executable:simple,1,5
destinationSpecification sourceSpecification
```

## Executando um script Ant e especificando códigos de retorno bem-sucedidos


O exemplo a seguir especifica um script Ant chamado `myscript` e transmite duas propriedades para o script: O script é executado usando o comando **fteAnt**. O valor para `successrc` é especificado como `>2&<7&!5|0|14`, que especifica que os códigos de retorno de 0, 3, 4, 6 e 14 indicam sucesso.

```
antscript:myscript(prop1=fred,prop2=bob),,,>2&<7&!5|0|14
```

Para executar este programa no agente de destino `AGENT2` antes do início da transferência, use o seguinte comando:

```
fteCreateTransfer -sa AGENT1 -da AGENT2 -predst
"antscript:myscript(prop1=fred,prop2=bob),,,>2&<7&!5|0|14"destinationSpecification sourceSpecification
```

## Executando um script Ant e especificando destinos para chamar

O exemplo a seguir especifica um script Ant chamado `script2` e dois destinos, `target1` e `target2`, a serem chamados. A propriedade `prop1` também é transmitida com um valor de `recmfm(F,B)`. Parênteses, vírgulas (,) e barras invertidas (\) são caracteres especiais em comandos MFT e devem ser escapados com um caractere de barra invertida (\).  Caminhos de arquivo no Windows podem ser especificados usando barras duplas invertidas (\\) como separador ou usando barras simples (/). No exemplo a seguir, a vírgula (,) e parênteses são escapados usando um caractere de barra invertida (\).

```
antscript:script2(target1,target2,prop1=recmfm\F\B\),,,>2&<7&!5|0|14
```

Para executar este programa no agente de destino `AGENT2` após a conclusão da transferência, use o seguinte comando:

```
fteCreateTransfer -sa AGENT1 -da AGENT2
-postdst "antscript:script2(target1,target2,prop1=recmfm\F\B\),,,>2&<7&!5|0|14"
destinationSpecification sourceSpecification
```

### Usando metadados em um script Ant

É possível especificar uma tarefa Ant como qualquer uma das chamadas a seguir para uma transferência:

- pré-origem
- pós-origem
- pré-destino
- pós-destino

Quando a tarefa Ant é executada, os metadados do usuário da transferência são disponibilizados usando variáveis de ambiente.. É possível acessar esses dados utilizando, por exemplo, o seguinte código:

```
<property environment="environment" />
<echo>${environment.mymetadata}</echo>
```

em que mymetadata é o nome de alguns metadados inseridos na transferência.

### Executando um Script de JCL

O exemplo a seguir especifica um script JCL chamado ZOSBATCH. Um valor de 3 é especificado para `retrycount`, um valor de 30 é especificado para `retrywait` e um valor de 0 é especificado para `successrc`. Estes valores significam que o script é tentado novamente três vezes caso não retorne um código de retorno bem-sucedido de 0, com uma espera de trinta segundos entre cada tentativa.

```
jcl:ZOSBATCH,3,30,0
```

em que ZOSBATCH é membro de um PDS chamado MYSYS.JCL e o arquivo `agent.properties` contém a linha `commandPath=...:/'MYSYS.JCL':...`

Para executar este programa no agente de origem AGENT1 após a conclusão da transferência, use o seguinte comando:

```
fteCreateTransfer -sa AGENT1 -da AGENT2 -postsrc jcl:ZOSBATCH,3,30,0
destinationSpecification sourceSpecification
```

### Tarefas relacionadas

Especificando programas para executar com o MFT

### Referências relacionadas

**fteCreateTransfer**: [iniciar uma nova transferência de arquivos](#)

## fteAnt: executar tarefas Ant no MFT

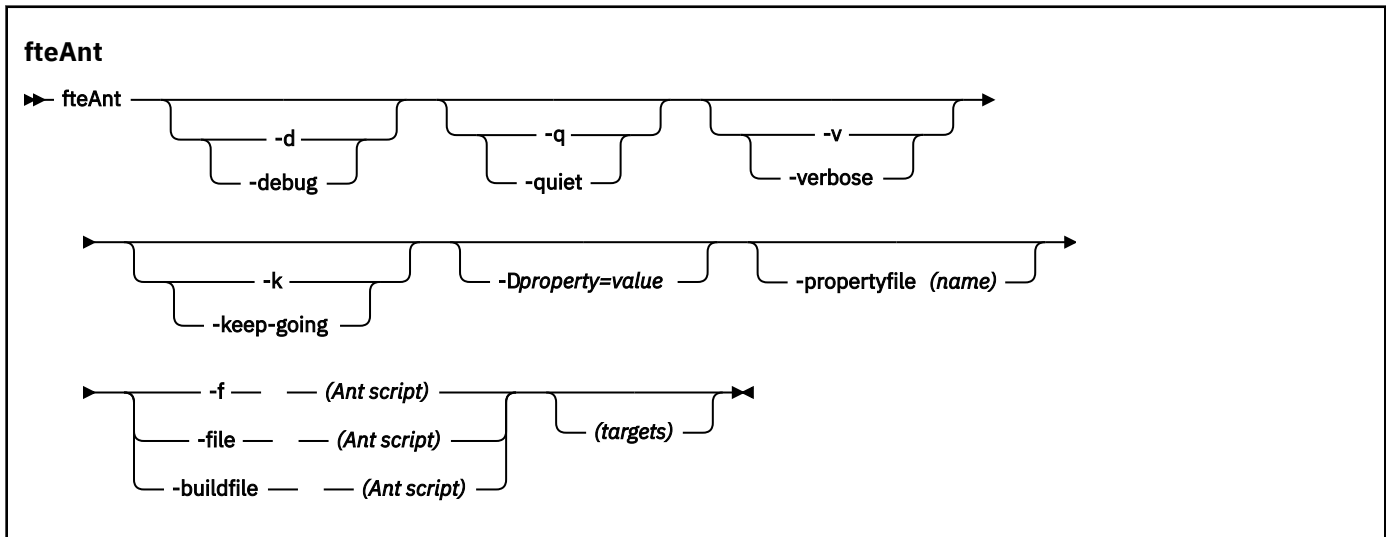
O comando **fteAnt** executa scripts Ant em um ambiente que possui tarefas do Managed File Transfer Ant disponíveis. Diferente do comando padrão **ant**, **fteAnt** requer a definição de um arquivo de script.

### tarefas do MFT Ant e parâmetros aninhados

O Managed File Transfer fornece várias tarefas do Ant que podem ser usadas para acessar os recursos de transferência de arquivos. Há também um conjunto de parâmetros aninhados disponíveis; esses parâmetros descrevem conjuntos aninhados de elementos que são comuns em várias das tarefas Ant fornecidas

A sintaxe do comando **fteAnt**, parâmetros, exemplo de uso e códigos de retorno são descritos no restante deste tópico. Para obter detalhes das tarefas do Ant e parâmetros aninhados que são fornecidos pelo MFT, ver os subtópicos.

## Sintaxe fteAnt



## Parâmetros

### **-debug ou -d**

Opcional. Gera a saída de depuração.

### **-quiet ou -q**

Opcional. Gera a saída mínima.

### **-verbose ou -v**

Opcional. Gera a saída detalhada.

### **-keep-going ou -k**

Opcional. Execute todos os destinos que não dependem de destinos falhos.

### **-D property=value**

Opcional. Use *value* para uma *property* especificada. As propriedades configuradas com **-D** têm precedência sobre as configuradas em um arquivo de propriedades.

Use a propriedade **com.ibm.wmqfte.propertyset** para especificar o conjunto de opções de configuração que são usadas para tarefas do Ant Use o nome de um gerenciador de filas de coordenação não padrão como o valor para esta propriedade. Em seguida, as tarefas do Ant usam o conjunto de opções de configuração associadas a esse gerenciador de filas de coordenação não padrão. Se você não especificar essa propriedade, o conjunto padrão de opções de configuração baseado no gerenciador de filas de coordenação padrão será usado. Se você especificar o atributo **cmdqm** para uma tarefa Ant, esse atributo terá precedência sobre o conjunto de opções de configuração especificadas para o comando **fteAnt**. Este comportamento é aplicável independentemente de você estar usando o conjunto padrão de opções de configuração ou especificando um conjunto com a propriedade **com.ibm.wmqfte.propertyset**.

### **-propertyfile (name)**

Opcional. Carrega todas as propriedades de um arquivo com as propriedades **-D** com precedência.

### **-f (Ant script), -file (Ant script) ou -buildfile (Ant script)**

Obrigatório. Especifica o nome do script Ant para executar.

### **Destinos do**

Opcional. O nome de um ou mais destinos para executar a partir do script Ant. Se você não especificar um valor para este parâmetro, o destino padrão para o script será executado.

### **-version**

Opcional. Exibe o comando Managed File Transfer e versões Ant .

### **-? ou -h**

Opcional. Exibe a sintaxe de comando.

### **exemplo**

Nesse exemplo, a cópia (**copy**) de destino no script Ant `fte_script.xml` é executada e o comando grava a saída de depuração na saída padrão.

```
fteAnt -d -f fte_script.xml copy
```

## **Códigos de retorno**

### **0**

O comando foi concluído com êxito.

### **1**

O comando foi concluído com falhas.

Outros códigos de retorno de status também podem ser especificados a partir de scripts Ant, por exemplo, usando a tarefa com falha Ant

Consulte [Falha](#) para obter mais informações..

### **Conceitos relacionados**

[Introdução ao uso de scripts Ant com MFT](#)

### **Tarefas relacionadas**

[Usando o Apache Ant com o MFT](#)

### **Referências relacionadas**

[Tarefas Ant de amostra para o MFT](#)

## **fte: tarefa Ant awaitoutcome**

Aguarda a conclusão de uma operação **fte:filecopy**, **fte:filemove** ou **fte:call**.

## **Atributos**

### **ID**

Obrigatório. Identifica a transferência da qual aguardar um resultado. Geralmente, esta é uma propriedade configurada pelo atributo `idProperty` das tarefas [fte:filecopy](#), [fte:filemove](#) ou [fte:call](#).

### **rcproperty**

Obrigatório. Nomeia uma propriedade na qual armazenar o código de retorno da tarefa **fte:awaitoutcome**.

### **timeout**

Opcional. A quantidade máxima de tempo, em segundos, para aguardar até a conclusão da operação. O tempo limite mínimo é um segundo. Se não for especificado um valor de tempo limite, a tarefa **fte:awaitoutcome** aguardará indefinidamente o resultado da operação a ser determinado.

### **exemplo**

Neste exemplo, uma cópia de arquivo é iniciada e seu identificador é armazenado na propriedade `copy.id`. Enquanto a cópia é processada, outros processos podem ocorrer. A instrução **fte:awaitoutcome** é usada para aguardar até a conclusão da operação de cópia. A instrução **fte:awaitoutcome** identifica qual operação aguardar para usar o identificador armazenado na propriedade `copy.id`. **fte:awaitoutcome** armazena um código de retorno que indica o resultado da operação de cópia em uma propriedade chamada `copy.result`.

```
<-- issue a file copy request -->
```



```

<fte:filecopy
src="AGENT1@QM1"
dst="AGENT2@QM2"
idproperty="copy.id"
outcome="defer">

<fte:filespec
srcfilespec="/home/fteuser1/file.bin"
dstdir="/home/fteuser2"/>

</fte:filecopy>

<fte:awaitoutcome id="{copy.id}" rcProperty="copy.rc"/>

<echo>Copy id="{copy.id}" rc="{copy.rc}"</echo>

```

## Tarefas relacionadas

Usando o Apache Ant com o MFT

## fte: chamar tarefa Ant

É possível usar a tarefa **fte:call** para chamar scripts e programas remotamente.

Esta tarefa permite enviar uma solicitação **fte:call** para um agente. O agente processa esse pedido executando um script ou programa e retornando o resultado. Os comandos a serem chamados devem ser acessíveis ao agente. Assegure-se de que o valor da propriedade `commandPath` no arquivo `agent.properties` inclua o local dos comandos a serem chamados. Qualquer informação de caminho especificada pelo elemento aninhado no comando deve ser relativa aos locais especificados pela propriedade `commandPath`. Por padrão, `commandPath` é vazio para que o agente não possa chamar nenhum comando. Para obter mais informações sobre esta propriedade, consulte [propriedadecommandPath MFT](#).

Para obter mais informações sobre o arquivo `agent.properties`, consulte [O arquivo MFTagent.properties](#).

## Atributos

### agent

Obrigatório. Especifica o agente para o qual enviar a solicitação **fte:call**. Especifique as informações do agente no formato: `agentname@qmgrname`, em que `agentname` é o nome do agente e `qmgrname` é o nome do gerenciador de filas ao qual este agente está diretamente conectado.

### cmdqm

Opcional. O gerenciador de filas de comando para o qual enviar a solicitação. Especifique essas informações no formulário `qmgrname@host@port@channel`, em que:

- `qmgrname` é o nome do gerenciador de filas
- `host` é o nome do host opcional do sistema onde o gerenciador de filas está executando
- `port` é o número de porta opcional que o gerenciador de filas está escutando
- `channel` é o canal SVRCONN opcional a ser usado

Se você omitir as informações de `host`, `port` ou `channel` para o gerenciador de filas de comando, as informações de conexão especificadas no arquivo `command.properties` serão usadas.



**Atenção:** Se nenhum valor for especificado para:

- Variável `host`, o modo de ligações é usado
- Variável `port`, o valor 1414 é usado
- `canal` variável, o `SYSTEM.DEF.SVRCONN` é usado..

Consulte [O arquivo command.properties do MFT](#) para obter mais informações.

No entanto, não é possível ignorar os atributos no meio, por exemplo, `qmgrname@host@@channel`. É possível ter, por exemplo, `qmgrname@host` ou `qmgrname@host@port` ou `qmgrname@hostport@@channel`.

O MFT divide o atributo fornecido usando o delimitador @ Dependendo do número de tokens localizados, ele usa o primeiro token como *qmgrname*, o segundo como *host*, o terceiro como *port* e, finalmente, *channel*.

Para obter mais informações, consulte [O arquivo MFT command.properties](#)

É possível usar a propriedade **com.ibm.wmqfte.propertySet** para especificar qual arquivo `command.properties` deve ser usado. Para obter mais informações, consulte [com.ibm.wmqfte.propertySet](#).

Se você não usar o atributo `cmdqm`, a tarefa será padronizada para usar a propriedade `com.ibm.wmqfte.ant.commandQueueManager`, se ela estiver configurada. Se a propriedade `com.ibm.wmqfte.ant.commandQueueManager` não for configurada, será tentada uma conexão com o gerenciador de filas padrão, definida no arquivo `command.properties`. O formato da propriedade `com.ibm.wmqfte.ant.commandQueueManager` é igual ao atributo `cmdqm`, ou seja, `qmgrname@host@port@channel`.

### **idproperty**

Opcional, a menos que você tenha especificado `outcome` como `defer`. Especifica o nome de uma propriedade ao qual designar o identificador de transferência. Identificadores de transferência são gerados no momento que um pedido de transferência é enviado e é possível usar identificadores de transferência para rastrear o andamento de uma transferência, diagnosticar problemas com uma transferência e cancelar uma transferência.

Você não pode especificar essa propriedade se já tiver especificado uma propriedade `outcome` como `ignore`. No entanto, deve-se especificar `idproperty` se já tiver especificado uma propriedade `outcome` como `defer`.

### **jobname**

Opcional. Designa um nome da tarefa à solicitação **fte:call**. É possível usar nomes de tarefas para criar grupos lógicos de transferências. Use a tarefa “[fte: uuid Ant tarefa](#)” na página 2186 para gerar nomes de tarefas com pseudônimos exclusivos. Se você não usar o atributo `jobname`, a tarefa será padronizada para usar o valor de propriedade `com.ibm.wmqfte.ant.jobName`, se essa propriedade for configurada. Se você não configurar esta propriedade, nenhum nome da tarefa será associado à solicitação **fte:call**.

### **origuser**

Opcional. Especifica o ID do usuário original a ser associado à solicitação **fte:call**. Se você não usar o atributo `origuser`, a tarefa será padronizada para usar o ID de usuário que é usado para executar o script Ant.

### **outcome**

Opcional. Determina se a tarefa espera que a operação **fte:call** seja concluída antes de retornar o controle para o script Ant Especifique uma das seguintes opções:

#### **await**

A tarefa aguarda a conclusão da operação **fte:call** antes de retornar. Quando um `outcome` de `await` for especificado, o atributo `idproperty` será opcional.

#### **defer**

A tarefa retorna assim que a solicitação **fte:call** tiver sido enviada e considera que o resultado da operação de chamada será tratado posteriormente usando as tarefas `awaitoutcome` ou `ignoreoutcome`. Quando um `outcome` de `defer` for especificado, o atributo `idproperty` será obrigatório.

#### **ignore**

Se o resultado da operação **fte:call** não for importante, será possível especificar um valor de `ignore`. A tarefa então retorna assim que a solicitação **fte:call** tiver sido enviada, sem alocar recursos para rastrear o resultado do comando. Quando um `outcome` de `ignore` for especificado, o atributo `idproperty` não poderá ser especificado.

Se você não especificar o atributo `outcome`, a tarefa será padronizada para usar o valor `await`.

### **rcproperty**

Opcional. Especifica o nome de uma propriedade à qual designar o resultado da solicitação **fte:call**. O código de resultado reflete o resultado geral da solicitação **fte:call**.

Você não pode especificar essa propriedade se também tiver especificado uma propriedade `outcome` como `ignore` ou `defer`. Entretanto, deve-se especificar `rcproperty` se já tiver especificado um resultado `await`.

## Parâmetros especificados como elementos aninhados

### **fte:command**

Especifica o comando a ser chamado pelo agente. É possível associar apenas um único elemento `fte:command` a uma determinada operação **fte:call**. O comando a ser chamado deve estar localizado no caminho especificado pela propriedade `commandPath` no arquivo `agent.properties` do agente.

### **fte:metadata**

É possível especificar os metadados para associar com a operação de chamada. Esses metadados são gravados nas mensagens de log geradas pela operação de chamada. Você só pode associar um único bloco de metadados com um elemento de transferência específico; no entanto esse bloco pode conter várias partes de metadados.

### **exemplo**

Este exemplo mostra como chamar um comando em AGENT1, em execução em um gerenciador de filas QM1. O comando a ser chamado é o script `command.sh` o script será chamado com um único argumento `xyz`. O comando `command.sh` está localizado no caminho especificado pela propriedade `commandPath` no arquivo `agent.properties` do agente.

```
<fte:call cmdqm="QM0@localhost@1414@SYSTEM.DEF.SVRCONN"
  agent="AGENT1@QM1"
  rcproperty="call.rc"
  origuser="bob"
  jobname="{\$!job.id}">

  <fte:command command="command.sh" successrc="1" retrycount="5" retrywait="30">
    <fte:arg value="xyz" />
  </fte:command>

  <fte:metadata>
    <fte:entry name="org.foo.accountName" value="BDG3R" />
  </fte:metadata>

</fte:call>
```

### **Tarefas relacionadas**

Usando o Apache Ant com o MFT

## **fte: cancelar tarefa Ant**

Cancela uma transferência gerenciada ou uma chamada gerenciada do Managed File Transfer. Uma transferência gerenciada pode ter sido criada usando as tarefas **fte:filecopy** ou **fte:filemove**. Uma chamada gerenciada pode ter sido criada usando a tarefa **fte:call**.

## **Atributos**

### **agent**

Obrigatório. Especifica o agente para o qual enviar a solicitação **fte:cancel**. O valor está no formato: `agentname@qmgrname`, em que `agentname` é o nome do agente e `qmgrname` é o nome do gerenciador de filas ao qual este agente está diretamente conectado.

### **cmdqm**

Opcional. O gerenciador de filas de comando para o qual enviar a solicitação. Especifique essas informações no formulário `qmgrname@host@port@channel`, em que:

- `qmgrname` é o nome do gerenciador de filas
- `host` é o nome do host opcional do sistema onde o gerenciador de filas está executando

- *port* é o número de porta opcional que o gerenciador de filas está escutando
- *channel* é o canal SVRCONN opcional a ser usado

Se você omitir as informações de *host*, *port* ou *channel* para o gerenciador de filas de comando, as informações de conexão especificadas no arquivo `command.properties` serão usadas.



**Atenção:** Se nenhum valor for especificado para:

- Variável *host*, o modo de ligações é usado
- Variável *port*, o valor 1414 é usado
- *canal* variável, o SYSTEM.DEF.SVRCONN é usado..

Consulte [O arquivo `command.properties` do MFT](#) para obter mais informações.

No entanto, não é possível ignorar os atributos no meio, por exemplo, `qmgrname@host@@channel`. É possível ter, por exemplo, `qmgrname@hostou qmgrname@host@portou qmgrname@hostport@@channel`.

O MFT divide o atributo fornecido usando o delimitador @ Dependendo do número de tokens localizados, ele usa o primeiro token como *qmgrname*, o segundo como *host*, o terceiro como *port* e, finalmente, *channel*.

Para obter mais informações, consulte [O arquivo MFT `command.properties`](#)

É possível usar a propriedade **`com.ibm.wmqfte.propertySet`** para especificar qual arquivo `command.properties` deve ser usado. Para obter mais informações, consulte [com.ibm.wmqfte.propertySet](#).

Se você não usar o atributo `cmdqm`, a tarefa será padronizada para usar a propriedade `com.ibm.wmqfte.ant.commandQueueManager`, se ela estiver configurada. Se a propriedade `com.ibm.wmqfte.ant.commandQueueManager` não for configurada, será tentada uma conexão com o gerenciador de filas padrão, definida no arquivo `command.properties`. O formato da propriedade `com.ibm.wmqfte.ant.commandQueueManager` é igual ao atributo `cmdqm`, ou seja, `qmgrname@host@port@channel`.

## ID

Obrigatório. Especifica o identificador de transferência de transfer para cancel. Os identificadores de transferência são gerados no ponto em que uma solicitação de transferência é enviada pelas tarefas `fte:filecopy` e `fte:filemove`.

## origuser

Opcional. Especifica o identificador de usuário original para associar com o pedido **`cancel`**. Se o atributo `origuser` não for usado, a tarefa será padronizada para usar o ID do usuário usado para executar o script Ant.

## exemplo

O exemplo envia uma solicitação **`fte:cancel`** para o gerenciador de filas de comando `qm0`. A solicitação **`fte:cancel`** é direcionada em `agent1` no gerenciador de filas `qm1` para o identificador de transferência preenchido pela variável `transfer.id`. O pedido é executado usando o ID do usuário "bob".

```
<fte:cancel cmdqm="qm0@localhost@1414@SYSTEM.DEF.SVRCONN"
  agent="agent1@qm1"
  id="${transfer.id}"
  origuser="bob"/>
```

## Tarefas relacionadas

Usando o Apache Ant com o MFT

## Tarefa `fte:filecopy` Ant

A tarefa **`fte:filecopy`** copia arquivos entre agentes do Managed File Transfer. O arquivo não é excluído do agente de origem.

## Atributos

### cmdqm

Opcional. O gerenciador de filas de comando para o qual enviar a solicitação. Especifique essas informações no formulário `qmgrname@host@port@channel`, em que:

- `qmgrname` é o nome do gerenciador de filas
- `host` é o nome do host opcional do sistema onde o gerenciador de filas está executando
- `port` é o número de porta opcional que o gerenciador de filas está escutando
- `channel` é o canal SVRCONN opcional a ser usado

Se você omitir as informações de `host`, `port` ou `channel` para o gerenciador de filas de comando, as informações de conexão especificadas no arquivo `command.properties` serão usadas.



**Atenção:** Se nenhum valor for especificado para:

- Variável `host`, o modo de ligações é usado
- Variável `port`, o valor 1414 é usado
- `canal` variável, o `SYSTEM.DEF.SVRCONN` é usado..

Consulte [O arquivo `command.properties` do MFT](#) para obter mais informações.

No entanto, não é possível ignorar os atributos no meio, por exemplo, `qmgrname@host@@channel`. É possível ter, por exemplo, `qmgrname@host` ou `qmgrname@host@port` ou `qmgrname@hostport@@channel`.

O MFT divide o atributo fornecido usando o delimitador `@`. Dependendo do número de tokens localizados, ele usa o primeiro token como `qmgrname`, o segundo como `host`, o terceiro como `port` e, finalmente, `channel`.

Para obter mais informações, consulte [O arquivo `MFT command.properties`](#)

É possível usar a propriedade `com.ibm.wmqfte.propertySet` para especificar qual arquivo `command.properties` deve ser usado. Para obter mais informações, consulte [com.ibm.wmqfte.propertySet](#).

Se você não usar o atributo `cmdqm`, a tarefa será padronizada para usar a propriedade `com.ibm.wmqfte.ant.commandQueueManager`, se ela estiver configurada. Se a propriedade `com.ibm.wmqfte.ant.commandQueueManager` não for configurada, será tentada uma conexão com o gerenciador de filas padrão, definida no arquivo `command.properties`. O formato da propriedade `com.ibm.wmqfte.ant.commandQueueManager` é igual ao atributo `cmdqm`, ou seja, `qmgrname@host@port@channel`.

### dst

Obrigatório. Especifica o agente de destino para a operação de cópia. Especifique estas informações no formato: `agentname@qmgrname`, em que `agentname` é o nome do agente de destino e `qmgrname` é o nome do gerenciador de filas ao qual esse agente é diretamente conectado.

### idproperty

Opcional, a menos que você tenha especificado `outcome` como `defer`. Especifica o nome de uma propriedade ao qual designar o identificador de transferência. Identificadores de transferência são gerados no momento que um pedido de transferência é enviado e é possível usar identificadores de transferência para rastrear o andamento de uma transferência, diagnosticar problemas com uma transferência e cancelar uma transferência.

Você não pode especificar essa propriedade se já tiver especificado uma propriedade `outcome` como `ignore`. No entanto, deve-se especificar `idproperty` se já tiver especificado uma propriedade `outcome` como `defer`.

### jobname

Opcional. Designa um nome de tarefa ao pedido de cópia. É possível usar nomes de tarefas para criar grupos lógicos de transferências. Use a tarefa “`fte: uuid Ant tarefa`” na [página 2186](#) para gerar nomes de tarefas com pseudônimos exclusivos. Se você não usar o atributo `jobname`, a tarefa será padronizada para usar o valor de propriedade `com.ibm.wmqfte.ant.jobName`, se essa

propriedade for configurada. Se você não definir essa propriedade, nenhum nome de tarefa será associado ao pedido de cópia.

#### **origuser**

Opcional. Especifica o identificador de usuário de origem a ser associado ao pedido de cópia. Se você não usar o atributo `origuser`, a tarefa será padronizada para usar o ID do usuário usado para executar o script Ant.

#### **outcome**

Opcional. Determina se a tarefa espera que a operação de cópia seja concluída antes de retornar o controle para o script Ant. Especifique uma das seguintes opções:

##### **await**

A tarefa aguarda até a conclusão da operação `copy` antes de retornar. Quando um `outcome` de `await` for especificado, o atributo `idproperty` será opcional.

##### **defer**

A tarefa retorna assim que a solicitação de cópia tiver sido enviada e assume que o resultado da operação de cópia será tratado posteriormente usando as tarefas `"fte: tarefa Ant awaitoutcome"` na página 2172 ou `"fte: ignoreoutcome Ant tarefa"` na página 2184. Quando um `outcome` de `defer` for especificado, o atributo `idproperty` será obrigatório.

##### **ignore**

Se o resultado da operação de cópia não for importante, você poderá especificar um valor `ignore`. A tarefa é então retornada assim que o pedido de cópia foi enviado, sem alocar nenhum recurso para o rastreamento do resultado da transferência. Quando um `outcome` de `ignore` for especificado, o atributo `idproperty` não poderá ser especificado.

Se você não especificar o atributo `outcome`, a tarefa será padronizada para usar o valor `await`.

#### **priority**

Opcional. Especifica a prioridade para associar com o pedido `copy`. No geral, solicitações de transferência de prioridade mais alta têm precedência sobre solicitações de prioridade mais baixa. O valor da prioridade deve estar no intervalo de 0 - 9 (inclusive). Um valor de prioridade 0 é a prioridade mais baixa e um valor 9 é a prioridade mais alta. Se você não especificar o atributo `priority`, a transferência será padronizada com prioridade 0.

#### **rcproperty**

Opcional. Especifica o nome de uma propriedade à qual designar o código de resultado do pedido de cópia. O código de resultado reflete o resultado geral do pedido de cópia.

Você não pode especificar essa propriedade se também tiver especificado uma propriedade `outcome` como `ignore` ou `defer`. No entanto, deve-se especificar `rcproperty` se especificar um resultado de `await`.

#### **transferRecoveryTimeout**

Opcional. Configura o período de tempo, em segundos, durante o qual um agente de origem continuará tentando recuperar uma transferência de arquivos paralisada. Especifique uma das seguintes opções:

##### **-1**

O agente continuará tentando recuperar a transferência paralisada até que ela seja concluída. O uso dessa opção é equivalente ao comportamento padrão do agente quando a propriedade não está configurada.

##### **0**

O agente para a transferência de arquivos assim que ela entra em recuperação.

##### **>0**

O agente continuará tentando recuperar a transferência paralisada durante o período de tempo em segundos, conforme configurado pelo valor de número inteiro positivo especificado. Por exemplo,

```
<fte:filecopy cmdqm="qm0@localhost@1414@SYSTEM.DEF.SVRCONN"
src="agent1@qm1" dst="agent2@qm2"
rcproperty="copy.result" transferRecoveryTimeout="21600">
```

```
<fte:filespec srcfilespec="/home/fteuser1/file.bin" dstfile="/home/fteuser2/
file.bin"/>
</fte:filecopy>
```

indica que o agente continuará tentando recuperar a transferência por 6 horas a partir de quando ela entrar em recuperação. O valor máximo para esse atributo é 999999999.

Especificar o valor de tempo limite de recuperação de transferência dessa maneira configura-o em uma base de transferência. Para configurar um valor global para todas as transferências em uma rede do Managed File Transfer, é possível incluir uma propriedade nas Propriedades de tempo limite de recuperação de transferência. Para obter mais informações, veja Opção de tempo limite para transferências em recuperação.

#### **src**

Obrigatório. Especifica o agente de origem para a operação copy. Especifique estas informações no formato: *agentname@qmgrname*, em que *agentname* é o nome do agente de origem e *qmgrname* é o nome do gerenciador de filas ao qual este agente está diretamente conectado.

## **Parâmetros especificados como elementos aninhados**

#### **fte:filespec**

Obrigatório. Deve-se especificar pelo menos uma especificação de arquivo que identifique os arquivos a serem copiados. É possível especificar mais de uma especificação de arquivo, se necessário. Consulte "fte: filespec Ant elemento aninhado" na página 2186 para obter mais informações.

#### **fte:metadata**

É possível especificar os metadados para associar com a operação de cópia. Esses metadados são transmitidos com a transferência e gravados nas mensagens de log geradas pela transferência. Você só pode associar um único bloco de metadados com um elemento de transferência específico; no entanto esse bloco pode conter várias partes de metadados. Consulte o tópico fte:metadata para obter informações adicionais.

#### **fte:presrc**

Especifica que uma chamada de programa seja feita no agente de origem antes do início da transferência. É possível associar apenas um único elemento `fte:presrc` a uma determinada transferência. Consulte o tópico Chamada de programa para obter informações adicionais.

#### **fte:predst**

Especifica que uma chamada de programa seja feita no agente de destino antes do início da transferência. É possível associar apenas um único elemento `fte:predst` a uma determinada transferência. Consulte o tópico Chamada de programa para obter informações adicionais.

#### **fte:postsrc**

Especifica que uma chamada de programa seja feita no agente de origem após a conclusão da transferência. É possível associar apenas um único elemento `fte:postsrc` a uma determinada transferência. Consulte o tópico Chamada de programa para obter informações adicionais.

#### **fte:postdst**

Especifica que uma chamada de programa seja feita no agente de destino após a conclusão da transferência. É possível associar apenas um único elemento `fte:postdst` a uma determinada transferência. Consulte o tópico Chamada de programa para obter informações adicionais.

Se `fte:presrc`, `fte:predst`, `fte:postsrc`, `fte:postdst` e as saídas não retornarem um status de êxito, as regras serão as seguintes na ordem especificada:

1. Execute as saídas de início da origem. Se as saídas de início da origem falharem, a transferência falha e nada mais é executado.
2. Execute a chamada pré-origem (quando presente). Se a chamada pré-origem falhar, a transferência falha e nada mais é executado.
3. Execute as saídas de início do destino. Se as saídas de início de destino falharem, a transferência falha e nada mais é executado.



4. Execute a chamada pré-destino (quando presente). Se a chamada pré-destino falhar, a transferência falha e nada mais é executado.
5. Execute as transferências do arquivo.
6. Execute as saídas de término de destino. Não há nenhum status de falha para estas saídas.
7. Se a transferência for bem-sucedida (se alguns arquivos forem transferidos com sucesso, será considerada bem-sucedida) execute a chamada pós-destino (se presente). Se a chamada pós-destino falhar, a transferência falhará.
8. Execute as saídas de término de origem. Não há nenhum status de falha para estas saídas.
9. Se a transferência for bem-sucedida, execute a chamada pós-origem (se presente). Se a chamada pós-origem falhar, a transferência falha.

## Exemplos

Este exemplo mostra uma transferência de arquivo básica entre agent1 e agent2. O comando para iniciar a transferência de arquivo é enviado para um gerenciador de filas chamado qm0, usando uma conexão de modo de transporte de cliente. O resultado da operação de transferência de arquivo é designado à propriedade chamada `copy.result`.

```
<fte:filecopy cmdqm="qm0@localhost@1414@SYSTEM.DEF.SVRCONN"
  src="agent1@qm1" dst="agent2@qm2"
  rcproperty="copy.result">
  <fte:filespec srcfilespec="/home/fteuser1/file.bin" dstfile="/home/fteuser2/file.bin"/>
</fte:filecopy>
```

Este exemplo mostra a mesma transferência de arquivos, mas com a inclusão de metadados e um início de programa para ocorrer no agente de origem após a conclusão da transferência.

```
<fte:filecopy cmdqm="qm0@localhost@1414@SYSTEM.DEF.SVRCONN"
  src="agent1@qm1" dst="agent2@qm2"
  rcproperty="copy.result">
  <fte:metadata>
    <fte:entry name="org.example.departId" value="ACCOUNTS"/>
    <fte:entry name="org.example.batchGroup" value="A1"/>
  </fte:metadata>
  <fte:filespec srcfilespec="/home/fteuser1/file.bin" dstfile="/home/fteuser2/file.bin"/>
  <fte:postsrc command="/home/fteuser2/scripts/post.sh" successsrc="1" >
    <fte:arg value="/home/fteuser2/file.bin"/>
  </fte:postsrc>
</fte:filecopy>
```

## Conceitos relacionados

[Opção de tempo limite para transferências de arquivos em recuperação](#)

## Tarefas relacionadas

[Usando o Apache Ant com o MFT](#)

## Tarefa fte:filemove Ant

A tarefa **fte:filemove** move arquivos entre agentes do Managed File Transfer. Quando um arquivo é transferido com êxito do agente de origem para o agente de destino, o arquivo é excluído do agente de origem.

## Atributos

### cmdqm

Opcional. O gerenciador de filas de comando para o qual enviar a solicitação. Especifique essas informações no formulário `qmgrname@host@port@channel`, em que:



- *qmgrname* é o nome do gerenciador de filas
- *host* é o nome do host opcional do sistema onde o gerenciador de filas está executando
- *port* é o número de porta opcional que o gerenciador de filas está escutando
- *channel* é o canal SVRCONN opcional a ser usado

Se você omitir as informações de *host*, *port* ou *channel* para o gerenciador de filas de comando, as informações de conexão especificadas no arquivo `command.properties` serão usadas.



**Atenção:** Se nenhum valor for especificado para:

- Variável *host*, o modo de ligações é usado
- Variável *port*, o valor 1414 é usado
- *canal* variável, o SYSTEM.DEF.SVRCONN é usado..

Consulte [O arquivo `command.properties` do MFT](#) para obter mais informações.

No entanto, não é possível ignorar os atributos no meio, por exemplo, `qmgrname@host@@channel`. É possível ter, por exemplo, `qmgrname@host` ou `qmgrname@host@port` ou `qmgrname@hostport@@channel`.

O MFT divide o atributo fornecido usando o delimitador @. Dependendo do número de tokens localizados, ele usa o primeiro token como *qmgrname*, o segundo como *host*, o terceiro como *port* e, finalmente, *channel*.

Para obter mais informações, consulte [O arquivo MFT `command.properties`](#)

É possível usar a propriedade **`com.ibm.wmqfte.propertySet`** para especificar qual arquivo `command.properties` deve ser usado. Para obter mais informações, consulte [com.ibm.wmqfte.propertySet](#).

Se você não usar o atributo `cmdqm`, a tarefa será padronizada para usar a propriedade `com.ibm.wmqfte.ant.commandQueueManager`, se ela estiver configurada. Se a propriedade `com.ibm.wmqfte.ant.commandQueueManager` não for configurada, será tentada uma conexão com o gerenciador de filas padrão, definida no arquivo `command.properties`. O formato da propriedade `com.ibm.wmqfte.ant.commandQueueManager` é igual ao atributo `cmdqm`, ou seja, `qmgrname@host@port@channel`.

### **dst**

Obrigatório. Especifica o agente de destino para a operação de cópia. Especifique estas informações no formato: *agentname@qmgrname*, em que *agentname* é o nome do agente de destino e *qmgrname* é o nome do gerenciador de filas ao qual este agente está diretamente conectado.

### **idproperty**

Opcional, a menos que você tenha especificado `outcome` como `defer`. Especifica o nome de uma propriedade ao qual designar o identificador de transferência. Identificadores de transferência são gerados no momento que um pedido de transferência é enviado e é possível usar identificadores de transferência para rastrear o andamento de uma transferência, diagnosticar problemas com uma transferência e cancelar uma transferência.

Você não pode especificar essa propriedade se já tiver especificado uma propriedade `outcome` como `ignore`. No entanto, deve-se especificar `idproperty` se já tiver especificado uma propriedade `outcome` como `defer`.

### **jobname**

Opcional. Designa um nome de tarefa ao pedido de movimentação. É possível usar nomes de tarefas para criar grupos lógicos de transferências. Use a tarefa `fte:uuid` para gerar nomes de tarefas pseudoexclusivos. Se você não usar o atributo `jobname`, a tarefa será padronizada para usar o valor de propriedade `com.ibm.wmqfte.ant.jobName`, se essa propriedade for configurada. Se você não definir essa propriedade, nenhum nome de tarefa será associado ao pedido de movimentação.

### **origuser**

Opcional. Especifica o identificador de usuário de origem a ser associado ao pedido de movimentação. Se você não usar o atributo `origuser`, a tarefa será padronizada para usar o ID do usuário usado para executar o script Ant.

## outcome

Opcional. Determina se a tarefa espera que a operação de movimentação seja concluída antes de retornar o controle para o script Ant. Especifique uma das seguintes opções:

### await

A tarefa aguarda até a conclusão da operação move antes de retornar. Quando um outcome de await for especificado, o atributo idproperty será opcional.

### defer

A tarefa retorna assim que a solicitação de movimento tiver sido enviada e considera que o resultado da operação de movimentação será tratado posteriormente usando a tarefa “fte: tarefa Ant awaitoutcome” na página 2172 ou “fte: ignoreoutcome Ant tarefa” na página 2184. Quando um outcome de defer for especificado, o atributo idproperty será obrigatório.

### ignore

Se o resultado da operação de movimentação não for importante, você poderá especificar um valor ignore. A tarefa é então retornada assim que o pedido de movimentação foi enviado, sem alocar nenhum recurso para o rastreamento do resultado da transferência. Quando um outcome de ignore for especificado, o atributo idproperty não poderá ser especificado.

Se você não especificar o atributo outcome, a tarefa será padronizada para usar o valor await.

## priority

Opcional. Especifica a prioridade para associar com o pedido move. No geral, solicitações de transferência de prioridade mais alta têm precedência sobre solicitações de prioridade mais baixa. O valor da prioridade deve estar no intervalo de 0 - 9 (inclusive). Um valor de prioridade 0 é a prioridade mais baixa e um valor 9 é a prioridade mais alta. Se você não especificar o atributo priority, a transferência será padronizada com prioridade 0.

## rcproperty

Opcional. Especifica o nome de uma propriedade à qual designar o código de resultado do pedido de movimentação. O código de resultado reflete o resultado geral do pedido de movimentação.

Você não pode especificar essa propriedade se também tiver especificado uma propriedade outcome como ignore ou defer. Entretanto, deve-se especificar rcproperty se já tiver especificado um resultado await.

## transferRecoveryTimeout

Opcional. Configura o período de tempo, em segundos, durante o qual um agente de origem continuará tentando recuperar uma transferência de arquivos paralisada. Especifique uma das seguintes opções:

### -1

O agente continuará tentando recuperar a transferência paralisada até que ela seja concluída. O uso dessa opção é equivalente ao comportamento padrão do agente quando a propriedade não está configurada.

### 0

O agente para a transferência de arquivos assim que ela entra em recuperação.

### >0

O agente continuará tentando recuperar a transferência paralisada durante o período de tempo em segundos, conforme configurado pelo valor de número inteiro positivo especificado. Por exemplo,

```
<fte:filemove cmdqm="qm0@localhost@1414@SYSTEM.DEF.SVRCONN"
  src=agent1@qm1 dst="agent2@qm2"
  rcproperty="move.result" transferRecoveryTimeout="21600">
  <fte:filespec srcfilespec="/home/fteuser1/file.bin" dstfile="/home/fteuser2/
file.bin"/>
</fte:filemove
```

indica que o agente continuará tentando recuperar a transferência por 6 horas a partir de quando ela entrar em recuperação. O valor máximo para esse atributo é 999999999.

Especificar o valor de tempo limite de recuperação de transferência dessa maneira configura-o em uma base de transferência. Para configurar um valor global para todas as transferências em uma rede do Managed File Transfer, é possível incluir uma propriedade nas Propriedades de tempo limite de recuperação de transferência. Para obter mais informações, veja Opção de tempo limite para transferências em recuperação.

#### **src**

Obrigatório. Especifica o agente de origem para a operação move. Especifique estas informações no formato: *agentname@qmgrname*, em que *agentname* é o nome do agente de origem e *qmgrname* é o nome do gerenciador de filas ao qual este agente está diretamente conectado.

## **Parâmetros especificados como elementos aninhados**

#### **fte:filespec**

Obrigatório. Deve-se especificar pelo menos uma especificação de arquivo que identifique os arquivos a serem movidos. É possível especificar mais de uma especificação de arquivo, se necessário. Consulte “fte: filespec Ant elemento aninhado” na página 2186 para obter mais informações.

#### **fte:metadata**

Opcional. É possível especificar metadados a serem associados à operação de movimentação de arquivo. Esses metadados são transmitidos com a transferência e gravados nas mensagens de log geradas pela transferência. Você só pode associar um único bloco de metadados com um elemento de transferência específico; no entanto esse bloco pode conter várias partes de metadados. Consulte o tópico fte:metadata para obter informações adicionais.

#### **fte:presrc**

Opcional. Especifica que uma chamada de programa seja feita no agente de origem antes do início da transferência. É possível associar apenas um único elemento *fte:presrc* a uma determinada transferência. Consulte o tópico Chamada de programa para obter informações adicionais.

#### **fte:predst**

Opcional. Especifica que uma chamada de programa seja feita no agente de destino antes do início da transferência. É possível associar apenas um único elemento *fte:predst* a uma determinada transferência. Consulte o tópico Chamada de programa para obter informações adicionais.

#### **fte:postsrc**

Opcional. Especifica que uma chamada de programa seja feita no agente de origem após a conclusão da transferência. É possível associar apenas um único elemento *fte:postsrc* a uma determinada transferência. Consulte o tópico Chamada de programa para obter informações adicionais.

#### **fte:postdst**

Opcional. Especifica que uma chamada de programa seja feita no agente de destino após a conclusão da transferência. É possível associar apenas um único elemento *fte:postdst* a uma determinada transferência. Consulte o tópico Chamada de programa para obter informações adicionais.

Se *fte:presrc*, *fte:predst*, *fte:postsrc*, *fte:postdst* e as saídas não retornarem um status de êxito, as regras serão as seguintes na ordem especificada:

1. Execute as saídas de início da origem. Se as saídas de início da origem falharem, a transferência falha e nada mais é executado.
2. Execute a chamada pré-origem (quando presente). Se a chamada pré-origem falhar, a transferência falha e nada mais é executado.
3. Execute as saídas de início do destino. Se as saídas de início de destino falharem, a transferência falha e nada mais é executado.
4. Execute a chamada pré-destino (quando presente). Se a chamada pré-destino falhar, a transferência falha e nada mais é executado.
5. Execute as transferências do arquivo.
6. Execute as saídas de término de destino. Não há nenhum status de falha para estas saídas.

7. Se a transferência for bem-sucedida (se alguns arquivos forem transferidos com sucesso, a transferência será considerada bem-sucedida), execute a chamada de pós-destino (se presente). Se a chamada pós-destino falhar, a transferência falhará.
8. Execute as saídas de término de origem. Não há nenhum status de falha para estas saídas.
9. Se a transferência for bem-sucedida, execute a chamada de pós-origem (se presente). Se a chamada pós-origem falhar, a transferência falha.

## Exemplos

Este exemplo mostra uma movimentação de arquivo básica entre agent1 e agent2. O comando para iniciar a movimentação de arquivo é enviado para um gerenciador de filas chamado qm0, usando uma conexão de modo de transporte de cliente. O resultado da operação de transferência de arquivo é designado à propriedade chamada `move.result`.

```
<fte:filemove cmdqm="qm0@localhost@1414@SYSTEM.DEF.SVRCONN"
  src="agent1@qm1" dst="agent2@qm2"
  rcproperty="move.result">

  <fte:filespec srcfilespec="/home/fteuser1/file.bin" dstfile="/home/fteuser2/file.bin"/>
</fte:filemove>
```

## Conceitos relacionados

[Opção de tempo limite para transferências de arquivos em recuperação](#)

## Tarefas relacionadas

[Usando o Apache Ant com o MFT](#)

## fte:ignoreoutcome Ant tarefa

Ignore o resultado de um comando **fte:filecopy**, **fte:filemove** ou **fte:call** .. Quando você especifica uma tarefa **fte:filecopy**, **fte:filemove** ou **fte:call** para ter um resultado de defer, a tarefa Ant aloca recursos para rastrear esse resultado. Se você não estiver mais interessado no resultado, poderá usar a tarefa **fte:ignoreoutcome** para liberar esses recursos.

## Atributos

### ID

Obrigatório. Identifica o resultado que não é mais de seu interesse. Normalmente, você especifica esse identificador usando uma propriedade definida usando o atributo `idproperty` da tarefa [“Tarefa fte:filecopy Ant”](#) na página 2176, [“Tarefa fte:filemove Ant”](#) na página 2180 ou [“fte: chamar tarefa Ant”](#) na página 2173.

### exemplo

Este exemplo mostra como é possível usar a tarefa `fte:ignoreoutcome` para liberar os recursos alocados para rastrear o resultado da tarefa [“Tarefa fte:filecopy Ant”](#) na página 2176 anterior.

```
<!-- issue a file copy request -->
<fte:filecopy cmdqm="qm1@localhost@1414@SYSTEM.DEF.SVRCONN"
  src="agent1@qm1" dst="agent1@qm1"
  idproperty="copy.id"
  outcome="defer"/>

<!-- do some other things -->

<!-- decide that the result of the copy is not interesting -->
<fte:ignoreoutcome id="{copy.id}"/>
```

## Tarefas relacionadas

[Usando o Apache Ant com o MFT](#)

## tarefa fte: ping Ant

Essa tarefa do IBM MQ Managed File Transfer Ant executa o pings de um agente para obter uma resposta e, portanto, determina se o agente é capaz de processar transferências

**Nota:** IBM WebSphere MQ File Transfer Edition (FTE) não é mais um produto suportado Para migrar do FTE para o componente Managed File Transfer no IBM MQ, consulte [Migrando o Managed File Transfer](#).

### Atributos

#### agent

Obrigatório. Especifica o agente para o qual enviar a solicitação **fte:ping**. O valor está no formato: *agentname@qmgrname*, em que *agentname* é o nome do agente e *qmgrname* é o nome do gerenciador de filas ao qual este agente está diretamente conectado.

#### cmdqm

Opcional. O gerenciador de filas de comando para o qual enviar a solicitação. Especifique essas informações no formulário *qmgrname@host@port@channel*, em que:

- *qmgrname* é o nome do gerenciador de filas
- *host* é o nome do host opcional do sistema onde o gerenciador de filas está executando
- *port* é o número de porta opcional que o gerenciador de filas está escutando
- *channel* é o canal SVRCONN opcional a ser usado

Se você omitir as informações de *host*, *port* ou *channel* para o gerenciador de filas de comando, as informações de conexão especificadas no arquivo `command.properties` serão usadas.



**Atenção:** Se nenhum valor for especificado para:

- Variável *host*, o modo de ligações é usado
- Variável *port*, o valor 1414 é usado
- *canal* variável, o SYSTEM.DEF.SVRCONN é usado..

Consulte [O arquivo command.properties do MFT](#) para obter mais informações.

No entanto, não é possível ignorar os atributos no meio, por exemplo, *qmgrname@host@@channel*. É possível ter, por exemplo, *qmgrname@host* ou *qmgrname@host@port* ou *qmgrname@hostport@@channel*.

O MFT divide o atributo fornecido usando o delimitador @ Dependendo do número de tokens localizados, ele usa o primeiro token como *qmgrname*, o segundo como *host*, o terceiro como *port* e, finalmente, *channel*.

Para obter mais informações, consulte [O arquivo MFT command.properties](#)

É possível usar a propriedade **com.ibm.wmqfte.propertySet** para especificar qual arquivo `command.properties` deve ser usado. Para obter mais informações, consulte [com.ibm.wmqfte.propertySet](#).

Se você não usar o atributo `cmdqm`, a tarefa será padronizada para usar a propriedade `com.ibm.wmqfte.ant.commandQueueManager`, se ela estiver configurada. Se a propriedade `com.ibm.wmqfte.ant.commandQueueManager` não for configurada, será tentada uma conexão com o gerenciador de filas padrão, definida no arquivo `command.properties`. O formato da propriedade `com.ibm.wmqfte.ant.commandQueueManager` é igual ao atributo `cmdqm`, ou seja, *qmgrname@host@port@channel*.

#### rcproperty

Obrigatório. Nomeia uma propriedade na qual armazenar o código de retorno da operação **ping**.

#### timeout

Opcional. O tempo máximo, em segundos, para que a tarefa aguarde a resposta do agente. O tempo limite mínimo é de zero segundos, no entanto, um tempo limite de menos um também pode ser especificado, de forma que o comando aguarde indefinidamente a resposta do agente. Se nenhum valor for especificado para `timeout`, o padrão será aguardar até 5 segundos a resposta do agente.

## exemplo

Este exemplo envia uma solicitação **fte:ping** para agent1 hospedado por qm1. A solicitação **fte:ping** aguarda 15 segundos a resposta do agente. O resultado da solicitação **fte:ping** é armazenado em uma propriedade chamada ping.rc.

```
<fte:ping agent="agent1@qm1" rcproperty="ping.rc" timeout="15"/>
```

## Códigos de retorno

**0**

O comando foi concluído com êxito.

**2**

O comando atingiu o tempo limite.

## Tarefas relacionadas

[Usando o Apache Ant com o MFT](#)

## fte: uuid Ant tarefa

Gera um identificador exclusivo pseudo-aleatório e o designa a uma determinada propriedade. Por exemplo, você pode usar esse identificador para gerar nomes de tarefas para outras operações de transferência de arquivo.

## Atributos

### comprimento

Obrigatório. O comprimento numérico do UUID a ser gerado. Esse comprimento não inclui o comprimento de nenhum prefixo, especificado pelo parâmetro **prefix**.

### propriedade

Obrigatório. O nome da propriedade para designar ao UUID gerado.

### prefix

Opcional. Um prefixo a ser incluído no UUID gerado. Este prefixo não é contado como parte do comprimento do UUID, conforme especificado pelo parâmetro **length**.

## exemplo

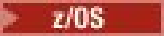

Este exemplo define um UUID que começa com as letras ABC, seguidas por 16 caracteres hexadecimais pseudo-aleatórios. O UUID é designado para uma propriedade chamada uuid.property.

```
<fte:uuid length="16" property="uuid.property" prefix="ABC"/>
```

## Tarefas relacionadas

[Usando o Apache Ant com o MFT](#)

## fte: filespec Ant elemento aninhado

O parâmetro **fte:filespec** é usado como um elemento aninhado em outras tarefas. Use **fte:filespec** para descrever um mapeamento entre um ou mais arquivos de origem, diretórios  ou conjuntos de dados e um destino. Normalmente, este elemento é usado ao expressar um conjunto de arquivos ou diretórios  ou conjuntos de dados para mover ou copiar.

## Aninhado por:

- A tarefa [fte:filecopy](#)

- A tarefa `fte:filemove`

## Atributos de Especificação de Origem

Deve-se especificar um de `srcfilespec` ou `srcqueue`.

### **srcfilespec**

Especifica a origem da operação do arquivo. O valor padrão desse atributo pode incluir um curinga.

### **srcqueue**

Especifica a origem da transferência como uma fila. A transferência move os dados a partir das mensagens armazenadas na fila especificada por esse atributo. Não será possível especificar este atributo se a tarefa **fte:filespec** estiver aninhada na tarefa **fte:filecopy**.

O atributo `srcqueue` não é suportado quando o agente de origem é um agente de ponte de protocolo.

## Atributos de Especificação de Destino

Deve-se especificar um de `dstdir`, `dstds`, `dstfilespace`, `dstfile`, `dstqueue` ou `dstpds`.

### **dstdir**

Especifica um diretório como o destino para uma operação de arquivo.

### **dstds**

Especifica um conjunto de dados como o destino para uma operação de arquivo.

Esse atributo é suportado somente quando o agente de destino está em execução na plataforma z/OS.

### **dstfile**

Especifica um arquivo como o destino para uma operação de arquivo.

### **dstfilespace**

Especifica um espaço no arquivo como o destino para uma operação de arquivo.

Esse atributo se aplica apenas se o agente de destino for um agente da Web IBM MQ 8.0 que tenha acesso ao espaço no arquivo de gateway da Web.

### **dstpds**

Especifica um conjunto de dados particionado como o destino para uma operação de arquivo.

Esse atributo é suportado somente quando o agente de destino está em execução na plataforma z/OS.

### **dstqueue**

Especifica uma fila como o destino para um arquivo na operação de mensagem. É possível, opcionalmente, incluir um nome de gerenciador de filas nessa especificação, usando o formato `QUEUE@QUEUEMANAGER`. Se você não especificar um nome de gerenciador de filas, o gerenciador de filas do agente de destino será usado, se você não tiver configurado a propriedade do agente `enableClusterQueueInputOutput` como `true`. Se a propriedade `enableClusterQueueInputOutput` for configurada como `true`, o agente de destino usará os procedimentos padrão do IBM MQ para determinar onde a fila está localizada. É necessário especificar um nome de fila válido que exista no gerenciador de filas.

Se você especificar o atributo `dstqueue`, não será possível especificar os atributos `srcqueue`, porque estes atributos são mutuamente exclusivos.

O atributo `dstqueue` não é suportado quando o agente de destino é um agente de ponte de protocolo.

## Atributos de Opção de Origem

### **srcencoding**

Opcional. A codificação do conjunto de caracteres usada pelo arquivo a ser transferido.



É possível especificar esse atributo apenas quando o atributo `conversion` é configurado para um valor de `text`.

Se você não especificar o atributo `srcencoding`, o conjunto de caracteres do sistema de origem é usado para transferências de texto.

### **srceol**

Opcional. O delimitador de final de linha usado pelo arquivo que está sendo transferido. Os valores válidos são os seguintes:

- `CRLF` - Use um caractere de retorno de linha seguido por um caractere de alimentação de linha como o delimitador de final de linha. Essa convenção é típica para sistemas Windows.
- `LF` - Use um caractere de feed de linha como o delimitador de fim de linha. Essa convenção é típica para sistemas UNIX.

É possível especificar este atributo apenas quando o atributo `conversion` for configurado para um valor `text`. Se você não especificar o atributo `srceol`, as transferências de texto automaticamente determinam o valor correto com base no sistema operacional do agente de origem.

### **z/OS** **srckeeptrailingspaces**

Opcional. Determina se os espaços à direita serão mantidos em registros de origem lidos a partir de um conjunto de dados de formato de comprimento fixo como parte de um modo de transferência de texto. Os valores válidos são os seguintes:

- `true` - os espaços à direita são mantidos.
- `false` - os espaços à direita são removidos.

Se você não especificar o atributo `srckeeptrailingspaces`, um valor padrão de `false` será especificado.

É possível especificar este atributo apenas se você também especificar o atributo `srcfilespec` e configurar o atributo `conversion` como um valor de `text`.

### **srcmsgdelimbytes**

Opcional. Especifica um ou mais valores de byte a ser inseridos como o delimitador ao anexar várias mensagens em um arquivo binário. Cada valor deve ser especificado como dois dígitos hexadecimais no intervalo de 00 a FF, prefixados por `x`. Diversos bytes devem ser separados por vírgula. Por exemplo, `srcmsgdelimbytes="x08,xA4"`. É possível especificar o atributo `srcmsgdelimbytes` apenas se você também especificou o atributo `srcqueue`. Não é possível especificar o atributo `srcmsgdelimbytes` se você também especificou o valor `text` para o atributo `conversion`.

### **srcmsgdelimtext**

Opcional. Especifica uma sequência de texto a ser inserida como o delimitador ao anexar várias mensagens em um arquivo de texto. É possível incluir sequências de escape do Java para Sequências literais no delimitador. Por exemplo, `srcmsgdelimtext="\u007d\n"`. O delimitador de texto é inserido após cada mensagem pelo agente de origem. O delimitador de texto é codificado para o formato binário usando a codificação de origem da transferência. Cada mensagem é lida em formato binário, o delimitador codificado é anexado no formato binário à mensagem e o resultado é transferido em formato binário para o agente de destino. Se a página de código do agente de origem incluir estados `shift-in` e `shift-out`, o agente supõe que cada mensagem está no estado `shift-out` no final da mensagem. No agente de destino os dados binários são convertidos da mesma forma como uma transferência de texto de arquivo para arquivo. É possível especificar o atributo `srcmsgdelimtext` se você também especificou o atributo `srcqueue` e um valor `text` para o atributo `conversion`.

### **srcmsgdelimposition**

Opcional. Especifica a posição à qual o delimitador de texto ou binário é inserido. Os valores válidos são os seguintes:

- `prefix` - os delimitadores são inseridos no arquivo de destino antes dos dados de cada mensagem.
- `postfix` - os delimitadores são inseridos no arquivo de destino após os dados de cada mensagem.

É possível especificar o atributo `srcmsgdelimposition` somente se também tiver especificado um dos atributos `srcmsgdelimbytes` ou `srcmsgdelimtext`.



### **srcmsggroups**

Opcional. Especifica se as mensagens são agrupadas por ID do grupo do IBM MQ. O primeiro grupo concluído é gravado no arquivo de destino. Se esse atributo não for especificado, todas as mensagens na fila de origem serão gravadas no arquivo de destino. É possível especificar o atributo `srcmsggroups` apenas se você também especificou o atributo `srcqueue`.

### **srcqueuetimeout**

Opcional. Especifica o tempo, em segundos, para aguardar até que uma das seguintes condições seja satisfeita:

- Para uma nova mensagem a ser gravada na fila.
- Se o atributo `srcmsggroups` foi especificado, para um grupo completo a ser gravado na fila.

Se nenhuma dessas condições for atendida dentro do tempo especificado pelo valor `srcqueuetimeout`, o agente de origem para a leitura da fila e conclui a transferência. Se o atributo `srcqueuetimeout` não for especificado, o agente de origem para a leitura da fila de origem imediatamente se a fila de origem estiver vazia ou, no caso em que o atributo `srcmsggroups` for especificado, se não existir nenhum grupo completo na fila. É possível especificar o atributo `srcqueuetimeout` apenas se você também especificou o atributo `srcqueue`.

Para obter informações sobre como configurar o valor `srcqueuetimeout`, consulte [Orientação para especificar um tempo de espera em uma transferência de mensagem para arquivo](#).

### **z/OS srcrcdelimbytes**

Opcional. Especifica um ou mais valores de byte a serem inseridos como o delimitador ao anexar diversos registros de um arquivo de origem orientado a registros em um arquivo binário. Você deve especificar cada valor como dois dígitos hexadecimais no intervalo 00-FF, prefixados por x. Diversos bytes devem ser separados por vírgula. Por exemplo:

```
srcrcdelimbytes="x08,xA4"
```

Será possível especificar o atributo `srcrcdelimbytes` somente se o arquivo de fonte de transferência for orientado a registros, por exemplo, um conjunto de dados do z/OS e o arquivo de destino for um arquivo normal não orientado a registros. Não é possível especificar o atributo `srcrcdelimbytes` se você também tiver especificado o valor `text` para o atributo `conversion`.

### **srcrcdelimpos**

Opcional. Especifica a posição em que o delimitador binário é inserido. Os valores válidos são os seguintes:

- prefix - os delimitadores são inseridos no arquivo de destino antes dos dados de cada registro de arquivo de origem orientado a registros.
- postfix - os delimitadores são inseridos no arquivo de destino após os dados de cada registro de arquivo de origem orientado a registros.

É possível especificar o atributo `srcrcdelimpos` apenas se você também tiver especificado o atributo `srcrcdelimbytes`.

## **Atributos de Opção de Destino**

### **dstencoding**

Opcional. A codificação do conjunto de caracteres a ser usada para o arquivo transferido.

É possível especificar esse atributo apenas quando o atributo `conversion` é configurado para um valor de `text`.

Se o atributo `dstencoding` não for especificado, o conjunto de caracteres do sistema de destino será usado para transferências de contexto.

### **dsteol**

Opcional. O delimitador de final de linha a ser usado para o arquivo transferido. Os valores válidos são os seguintes:

- CRLF - Use um caractere de retorno de linha seguido por um caractere de alimentação de linha como o delimitador de final de linha. Essa convenção é típica para sistemas Windows.
- LF - Use um caractere de feed de linha como o delimitador de fim de linha. Essa convenção é típica para sistemas UNIX.

É possível especificar esse atributo apenas quando o atributo `conversion` é configurado para um valor de `text`.

Se você não especificar o atributo `dstool`, as transferências de texto automaticamente determinam o valor correto com base no sistema operacional do agente de destino.

### **dstmsgdelimbytes**

Opcional. Especifica o delimitador hexadecimal a ser usado na divisão de um arquivo binário em várias mensagens. Todas as mensagens têm o mesmo ID do grupo do IBM MQ; a última mensagem no grupo tem a sinalização `LAST_MSG_IN_GROUP` do IBM MQ configurada. O formato para especificar um byte hexadecimal como um delimitador é `xNN`, em que `N` é um caractere no intervalo 0-9 ou a-f. É possível especificar uma sequência de bytes hexadecimais como um delimitador especificando uma lista de bytes hexadecimais separada por vírgulas, por exemplo: `x3e, x20, x20, xbf`.

É possível especificar o atributo `dstmsgdelimbytes` apenas se você também especificou o atributo `srcqueue` e a transferência estiver no modo binário. É possível especificar apenas um dos atributos `dstmsgsize`, `dstmsgdelimbytes` e `dstmsgdelimpattern`.

### **dstmsgdelimpattern**

Opcional. Especifica a expressão regular Java a ser usada na divisão de um arquivo de texto em múltiplas mensagens. Todas as mensagens têm o mesmo ID do grupo do IBM MQ; a última mensagem no grupo tem a sinalização `LAST_MSG_IN_GROUP` do IBM MQ configurada. O formato para especificar uma expressão regular como um delimitador é uma expressão regular entre parênteses, (*regular\_expression*), ou entre aspas duplas, "*regular\_expression*". Para obter mais informações, consulte [Expressões regulares usadas pelo MFT](#).

Por padrão, o comprimento de uma cadeia à qual a expressão regular pode corresponder é limitado pelo agente de destino a cinco caracteres. É possível mudar esse comportamento usando a propriedade do agente **maxDelimiterMatchLength**. Para obter mais informações, consulte [Propriedades do agente avançado do MFT](#).

É possível especificar o atributo `dstmsgdelimpattern` apenas se você também especificou o atributo `dstqueue` e a transferência estiver no modo texto. É possível especificar apenas um dos atributos `dstmsgsize`, `dstmsgdelimbytes` e `dstmsgdelimpattern`.

### **dstmsgdelimposition**

Opcional. Especifica a posição esperada do delimitador de texto ou binário. Os valores válidos são os seguintes:

- `prefix` - Os delimitadores são esperados no início de cada linha.
- `postfix` - Os delimitadores são esperados no final de cada linha.

É possível especificar o atributo `dstmsgdelimposition` somente se também tiver especificado o atributo `dstmsgdelimpattern`.

### **dstmsgincludedelim**

Opcional. Especifica se será incluído o delimitador usado para dividir o arquivo em várias mensagens nas mensagens. Se o atributo `dstmsgincludedelim` for especificado, o delimitador será incluído no final da mensagem que contiver os dados de arquivo que precedem o delimitador. Por padrão, o delimitador não é incluído nas mensagens. É possível especificar o atributo `dstmsgincludedelim` apenas se você também especificou um dos atributos `dstmsgdelimpattern` e `dstmsgdelimbytes`.

### **dstmsgpersist**

Opcional. Especifica se mensagens gravadas na fila de destino são persistentes. Os valores válidos são os seguintes:

- `true` - Grava mensagens persistentes na fila de destino. Esse é o valor-padrão.
- `false` - Grava mensagens não persistentes na fila de destino.

- `qdef` - O valor de persistência é obtido do atributo `DefPersistence` da fila de destino.

É possível especificar este atributo apenas quando o atributo `dstqueue` também for especificado.

### **dstmsgprops**

Opcional. Especifica se a primeira mensagem gravada na fila de destino pela transferência tem propriedades de mensagem do IBM MQ configuradas. Os valores possíveis são:

- `true` - Configura propriedades de mensagem na primeira mensagem criada pela transferência.
- `false` - Não configura propriedades de mensagem na primeira mensagem criada pela transferência. Esse é o valor-padrão.

Para obter mais informações, consulte as propriedades de mensagens [MQ configuradas pelo MFT em mensagens gravadas nas filas de destino](#)

É possível especificar este atributo apenas quando o atributo `dstqueue` também for especificado.

### **dstmsgsize**

Opcional. Especifica se o arquivo será dividido em várias mensagens com comprimento fixo. Todas as mensagens têm o mesmo ID do grupo do IBM MQ; a última mensagem no grupo tem a sinalização `LAST_MSG_IN_GROUP` do IBM MQ configurada. O tamanho das mensagens é especificado pelo valor de `dstmsgsize`. O formato de `dstmsgsize` é *lengthunits*, em que *length* é um valor de número inteiro positivo e *units* é um dos seguintes valores:

- B - Bytes. O valor mínimo permitido é duas vezes o valor máximo de bytes-por-caractere da página de código das mensagens de destino.
- K - Kibibytes. É o equivalente a 1024 bytes.
- M - Mebibytes. É o equivalente a 1024 kibibytes.

Se o arquivo for transferido em modo de texto e estiver em um conjunto de caracteres de byte duplo ou em um conjunto de caracteres multibyte, o arquivo será dividido em mensagens no limite de caractere mais próximo do tamanho de mensagem especificado.

É possível especificar o atributo `dstmsgsize` apenas se você também especificou o atributo `dstqueue`. É possível especificar apenas um dos atributos `dstmsgsize`, `dstmsgdelimbytes` e `dstmsgdelimpattern`.

### **dstunsupportedcodepage**

Opcional. Especifica a ação a ser executada se o gerenciador de filas de destino, conforme especificado pelo atributo `dstqueue`, não suportar a página de códigos usada quando transferir dados do arquivo para uma fila como uma transferência de texto. Os valores válidos para este atributo são os seguintes:

- `binary` - continuar a transferência, mas não aplicar a conversão de página de códigos nos dados que estão sendo transferidos. Especificar esse valor é equivalente a não configurar o atributo de conversão para `text`.
- `fail` - não continuar com a operação de transferência. O arquivo é registrado como possuindo falha de transferência. Esse é o padrão.

É possível especificar o atributo `dstunsupportedcodepage` apenas se você também tiver especificado o atributo `dstqueue` e um valor de `text` para o atributo `conversion`.

### **dsttruncaterecords**

Opcional. Especifica que os registros de destino maiores que o atributo do conjunto de dados `LRECL` serão truncados. Se configurado como `true`, os registros serão truncados. Se configurado como `false`, os registros serão agrupados. A configuração padrão é `false`. Este parâmetro é válido apenas para transferências de modo de texto nas quais o destino é um conjunto de dados.

## **Outros Atributos**

### **checksum**

Opcional. Determina o algoritmo usado para arquivos transferidos pela soma de verificação.

- MD5 - usar o algoritmo hash MD5.

- NONE - não usar um algoritmo de soma de verificação.

Se você não especificar o atributo checksum, um valor padrão MD5 é usado.

### Conversão

Opcional. Especifica o tipo de conversão para aplicar ao arquivo à medida que ele é transferido. Os valores possíveis são:

- binary - não aplica conversão.
- text - aplica conversão de página de código entre os sistemas de origem e destino. Também aplica conversão de delimitadores de linha. Os atributos srcencoding, dstencoding, srceol e dsteol influenciam a conversão que é aplicada.

Se você não especificar o atributo conversion, um valor padrão de binary será especificado.

### overwrite

Opcional. Determina se um arquivo de destino existente `z/OS` ou conjunto de dados pode ser substituído pela operação. Quando você especifica um valor de true, qualquer arquivo de destino `z/OS` ou conjuntos de dados é substituído. Quando você especifica um valor de false, a existência de um arquivo duplicado `z/OS` ou conjuntos de dados no destino resulta na falha da operação. Se o atributo overwrite não for especificado, um valor padrão false será especificado.

### recurse

Opcional. Determina se a transferência de arquivo será recursiva nos subdiretórios. Quando você especifica um valor true, a transferência é recursiva nos subdiretórios. Quando você especifica um valor false, a transferência não é recursiva nos subdiretórios. Se o atributo recurse não for especificado, um valor padrão false será especificado.

### exemplo

Esse exemplo especifica um fte:filespec com um arquivo de origem de file1.bin e um arquivo de destino de file2.bin.

```
<fte:filespec srcfilespec="/home/fteuser/file1.bin" dstfile="/home/fteuser/file2.bin"/>
```

### Tarefas relacionadas

Usando o Apache Ant com o MFT

## fte: metadados Ant elementos aninhados

O elemento metadata é usado para transmitir informações adicionais definidas pelo usuário com uma operação de transferência de arquivo.

Consulte “Metadados para saídas de usuário do MFT” na página 2196 para obter informações adicionais sobre como o Managed File Transfer usa metadados.

### Aninhado por:

- A tarefa [fte:filecopy](#)
- A tarefa [fte:filemove](#)
- A tarefa [fte:call](#)

## Parâmetros especificados como elementos aninhados

### fte:entry

Deve-se especificar pelo menos uma entrada dentro do elemento aninhado fte:metadata. É possível optar por especificar mais de uma entrada. Entradas associam um nome de chave a um valor. Chaves devem ser exclusivas em um bloco de fte:metadata

## Atributos de entrada

### nome

Obrigatório. O nome da chave que pertence a esta entrada. Este nome deve ser exclusivo em todos os parâmetros **entry** aninhados dentro de um elemento `fte:metadata`.

### valor

Obrigatório. O valor a ser designado para `entry`.

### exemplo

Este exemplo mostra uma definição de `fte:metadata` que contém duas entradas.

```
<fte:metadata>
  <fte:entry name="org.foo.partColor" value="red"/>
  <fte:entry name="org.foo.partSize" value="medium"/>
</fte:metadata>
```

### Tarefas relacionadas

Usando o [Apache Ant com o MFT](#)

## Elementos Aninhados de Chamada do Programa

Os programas podem ser iniciados usando um de cinco elementos aninhados: `fte:presrc`, `fte:predst`, `fte:postdst`, `fte:postsrc` e `fte:command`. Esses elementos aninhados instruem um agente a chamar um programa externo como parte de seu processamento. Antes de iniciar um programa, deve-se assegurar que o comando esteja em um local especificado pela propriedade `commandPath` no arquivo `agent.properties` do agente que executa o comando.

Mesmo que cada elemento de chamada do programa tenha um nome diferente, eles compartilham o mesmo conjunto de atributos e o mesmo conjunto de elementos aninhados. Os programas podem ser iniciados pelas tarefas Ant **`fte:filecopy`**, **`fte:filemove`** e **`fte:command`**.

Não é possível chamar programas a partir de um agente ponte `Connect:Direct`.

### Tarefas Ant que podem chamar programas:

- A tarefa `fte:filecopy` aninha parâmetros de chamada de programa usando os elementos aninhados `fte:predst`, `fte:postdst`, `fte:presrc` e `fte:postsrc`.
- A tarefa `fte:filemove` aninha os parâmetros de chamada do programa usando os elementos aninhado `fte:predst`, `fte:postdst`, `fte:presrc` e `fte:postsrc`.
- A tarefa `fte:call` aninha parâmetros de chamada do programa usando o elemento aninhado `fte:command`.

## Atributos

### comando

Obrigatório. Nomeia o programa a ser chamado. Para que o agente possa executar um comando, o comando deve estar em um local especificado pela propriedade `commandPath` no arquivo `agent.properties` do agente. Para obter mais informações, consulte [propriedadecommandPath MFT](#). Qualquer informação de caminho especificada no atributo `command` é considerada relativa a um local especificado pela propriedade `commandPath`. Quando `type` é `executable`, um programa executável é esperado; caso contrário, um script apropriado para o tipo de chamada é esperado.

### retrycount

Opcional. O número de vezes para tentar chamar novamente o programa se ele não retornar um código de retorno bem-sucedido. O programa nomeado pelo atributo `command` é chamado até este número de vezes. O valor designado a esse atributo não deve ser negativo. Se você não especificar o atributo `retrycount`, será usado um valor padrão de zero.

## retrywait

Opcional. O tempo de espera, em segundos, antes de tentar novamente a chamada de programa. Se o programa chamado pelo atributo `command` não retornar um código de retorno bem-sucedido e o atributo `retrycount` especificar um valor que não seja zero, esse parâmetro determinará o tempo de espera entre novas tentativas. O valor designado a esse atributo não deve ser negativo. Se você não especificar o atributo `retrywait`, um valor padrão zero será usado.

## successrc

Opcional. O valor desse atributo é usado para determinar quando a chamada do programa é executada com êxito. O código de retorno do processo para o comando é avaliado usando essa expressão. O valor pode ser composto de uma ou mais expressões combinadas com um caractere de barra vertical (|) para significar Booleano OR ou um e comercial (&) para significar Booleano AND. Cada expressão pode ser um dos seguintes tipos de expressão:

- um número para indicar um teste de igualdade entre o código de retorno e o número do processo.
- Um número prefixado com um caractere ">" para indicar um teste de maior que entre o número e o código de retorno do processo.
- Um número prefixado com um caractere "<" para indicar um teste inferior entre o número e o código de retorno do processo.
- Um número prefixado com um caractere "!=" para indicar um teste de não igualdade entre o número e o código de retorno do processo.

Por exemplo: `>2&<7&!5|0|14` é interpretado como o êxito dos seguintes códigos de retorno: 0, 3, 4, 6, 14. Todos os outros códigos de retorno são interpretados como malsucedidos. Se você não especificar o atributo `successrc`, será usado um valor padrão de zero. Isso significa que o comando de julgado como bem-sucedido se e apenas se, retornar um código zero.

## Tipo

Opcional. O valor desse atributo especifica que tipo de programa está sendo chamado. Especifique uma das seguintes opções:

### executable

A tarefa chama um programa executável. Pode ter argumentos adicionais especificados usando o elemento aninhado `arg`. Espera-se que o programa seja acessível no `commandPath` e, quando aplicável, tenha permissão de execução definida. Os scripts UNIX podem ser chamados desde que eles especifiquem um programa shell (por exemplo, a primeira linha do arquivo shell script é: `#!/bin/sh`). A saída de comando gravada para `stderr` ou `stdout` é enviada para o log Managed File Transfer para a chamada. No entanto, a quantidade de saída de dados é limitada pela configuração do agente. O padrão é 10K bytes de dados, mas você pode substituir esse padrão usando a propriedade do agente: `maxCommandOutput`.

### antscript

A tarefa executa o script Ant especificado, usando o comando `fteAnt`. As propriedades podem ser especificadas usando o elemento aninhado `property`. Destinos Ant podem ser especificados usando o elemento aninhado `target`. O script Ant deve estar acessível no `commandPath`. A saída Ant gravada para `stderr` ou `stdout` é enviada para o log Managed File Transfer para a chamada. No entanto, a quantidade de saída de dados é limitada pela configuração do agente. O padrão é 10K bytes de dados, mas você pode substituir esse padrão usando a propriedade do agente: `maxCommandOutput`.

### jcl

O valor `jcl` é suportado somente no z/OS e executa o script JCL do z/OS especificado. A JCL é submetida como uma tarefa e requer a presença do cartão de tarefa. Quando a tarefa for submetida com êxito, a saída de comando de JCL, gravada no log do Managed File Transfer, conterá o seguinte texto: `JOB job_name(job_id)`, em que:

- `job_name` é o nome da tarefa identificada pelo cartão de tarefa na JCL.
- `job_id` é o ID da tarefa gerado pelo sistema z/OS.

Se a tarefa não puder ser enviada com êxito, o comando de script JCL falhará e gravará uma mensagem no log indicando a razão da falha (por exemplo, nenhum cartão de tarefa está

presente). Para entender se a tarefa foi executada ou concluída com êxito, use um serviço do sistema como SDSF. O Managed File Transfer não fornece as informações porque ele submete apenas a tarefa; o sistema então determina quando executar a tarefa e como a saída da tarefa é apresentada. Como um script JCL é enviado como uma tarefa em lote, não é recomendável especificar `jc1` para um elemento aninhado `presrc` ou `predst`, porque você apenas sabe que a tarefa foi enviada com êxito e não sabe se ela foi executada com êxito até a conclusão antes do início da transferência. Não há elementos aninhados válidos com um tipo `jc1`.

O exemplo a seguir mostra uma tarefa JCL:

```
//MYJOB JOB
//*
//MYJOB EXEC PGM=IEBGENER
//SYSPRINT DD SYSOUT=H
//SYSUT1 DD DSN=FRED.DEMO.TXT,DISP=SHR
//SYSUT2 DD DSN=BOB.DEMO.TXT,DISP=(NEW,CATLG),
// RECFM=VB,LRECL=133,BLKSIZE=2048,
// SPACE=(TRK,(30,5),RLSE)
//SYSIN DD DUMMY
```

## Parâmetros especificados como elementos aninhados

### **fte:arg**

Válido somente quando o valor do atributo `type` é `executable`. Use elementos aninhados `fte:arg` para especificar argumentos para o programa que está sendo chamado como parte da chamada de programa. Os argumentos de programa são construídos a partir dos valores especificados pelos elementos `fte:arg` na ordem em que os elementos `fte:arg` são encontrados. É possível optar por especificar zero ou mais elementos `fte:arg` como elementos aninhados de uma chamada de programa.

### **fte:propriedade**

Válido somente quando o valor do atributo `type` é `antscript`. Use os atributos `name` e `value` dos elementos `fte:property` aninhados para transmitir pares nome-valor para o script Ant. É possível optar por especificar zero ou mais elementos `fte:property` como elementos aninhados de uma chamada de programa.

### **fte:target**

Válido somente quando o valor do atributo `type` é `antscript`. Especifique um destino no script Ant a ser chamado. É possível optar por especificar zero ou mais elementos `fte:target` como elementos aninhados de uma chamada de programa.

## Atributos Arg

### **Value**

Obrigatório. O valor do argumento para transmitir para o programa que está sendo chamado.

## Atributos de propriedade

### **nome**

Obrigatório. O nome de uma propriedade a ser transmitida para o script Ant

### **Value**

Obrigatório. O valor a ser associado ao nome da propriedade sendo passado para o script Ant.

## Exemplos

Este exemplo mostra uma chamada de programa `fte:postsrc` sendo especificada como parte de uma tarefa `fte:filecopy`. A chamada de programa é para um programa chamado `post.sh` e um único argumento de `/home/fteuser2/file.bin` é fornecido

```
<fte:filecopy
cmdqm="qm0@localhost@1414@SYSTEM.DEF.SVRCONN"
src="agent1@qm1" dst="agent2@qm2"
```



```

rcproperty="copy.result">
<fte:filespec srcfilespec="/home/fteuser1/file.bin" dstfile="/home/fteuser2/file.bin"/>
<fte:postsrc command="post.sh" successsrc="1" >
  <fte:arg value="/home/fteuser2/file.bin"/>
</fte:postsrc>
</fte:filecopy>

```

Este exemplo mostra uma chamada de programa `fte:command` sendo especificada como parte de uma tarefa `fte:call`. A chamada de programa é para um executável chamado `command.sh`, para o qual não é transmitido nenhum argumento de linha de comandos. Se `command.sh` não retornar um código de retorno de êxito de 1, o comando será tentado novamente após 30 segundos.

```

<fte:call
cmdqm="qm0@localhost@1414@SYSTEM.DEF.SVRCONN"
agent="agent1@qm1"
rcproperty="call.rc"
origuser="bob"
jobname="{job.id}">
<fte:command command="command.sh" successsrc="1" retrycount="5" retrywait="30"/>
</fte:call>

```

Este exemplo mostra uma chamada de programa `fte:command` sendo especificada como parte de uma tarefa `fte:call`. A chamada do programa é para os destinos de cópia e compactação em um script Ant chamado `script.xml`, que passou duas propriedades.

```

<fte:call
cmdqm="qm0@localhost@1414@SYSTEM.DEF.SVRCONN"
agent="agent1@qm1"
rcproperty="call.rc"
origuser="bob"
jobname="{job.id}">
<fte:command command="script.xml" type="antscript">
  <property name="src" value="AGENT5@QM5"/>
  <property name="dst" value="AGENT3@QM3"/>
  <target name="copy"/>
  <target name="compress"/>
</fte:command>
</fte:call>

```

### Tarefas relacionadas

[Especificando programas para executar com o MFT](#)

[Usando o Apache Ant com o MFT](#)

## MFT saídas de usuário para referência de customização

Informações de referência para ajudar a configurar saídas de usuário para Managed File Transfer.

### Conceitos relacionados

[Saídas de usuário de origem e destino do MFT](#)

### Metadados para saídas de usuário do MFT

Há três tipos diferentes de metadados que podem ser fornecidos para rotinas de saída do usuário para o Managed File Transfer: metadados de ambiente, de transferência e de arquivo. Esses metadados são apresentados como mapas de pares de valores e chave Java.

### Metadados do Ambiente

Os metadados do ambiente são passados para todas as rotinas de saída de usuários e descrevem o ambiente do tempo de execução do agente a partir do qual a rotina de saída de usuário está sendo chamada. São de somente leitura e não podem ser atualizados por nenhuma rotina de saída do usuário.



<i>Tabela 882. Metadados do Ambiente</i>	
<b>Chave</b>	<b>Descrição</b>
AGENT_CONFIGURATION_DIRECTORY_KEY	O nome do diretório que contém as informações de configuração do agente.
AGENT_PRODUCT_DIRECTORY_KEY	O nome do diretório no qual o código do agente foi instalado.
AGENT_VERSION_KEY	Número da versão do tempo de execução do agente que chama a rotina de saída.

Os nomes da chave e os nomes do valor fornecidos na Tabela 1 são constantes definidas na interface EnvironmentMetaDataConstants.

### **Metadados de Transferência**

Os metadados de transferência são passados para todas as rotinas de saída de usuário. Os metadados consistem em valores fornecidos pelo sistema e valores fornecidos pelo usuário. Se você alterar quaisquer valores fornecidos pelo sistema, essas mudanças serão ignoradas. Os valores iniciais fornecidos pelo usuário para a saída de usuário no início da transferência de origem se baseiam nos valores que você forneceu ao definir a transferência. O agente de origem pode alterar valores fornecidos pelo usuário como parte do processamento da saída de usuário no início da transferência de origem. Essa saída de usuário é chamada antes que a transferência do arquivo inteiro inicie. As alterações são usadas nas chamadas subsequentes para outras rotinas de saída relacionadas a tal transferência. Os metadados de transferência são aplicados a uma transferência inteira.

Embora todas as saídas de usuários possam ler os valores dos metadados de transferência, apenas a saída de usuário no início da transferência de origem poderá alterá-los.

Não é possível usar metadados de transferência para propagar informações entre diferentes transferências de arquivos.

Os metadados de transferência fornecidos pelo sistema são detalhados na Tabela 2:

<i>Tabela 883. Metadados de Transferência</i>	
<b>Chave</b>	<b>Descrição</b>
DESTINATION_AGENT_KEY	O nome do agente que é o destino para a transferência.
JOB_NAME_KEY	O nome da tarefa associada com o pedido de transferência
MQMD_USER_KEY	O campo do usuário MQMD da mensagem usada para enviar o pedido de transferência
ORIGINATING_HOST_KEY	O nome do host especificado como o nome do host de origem no pedido de transferência
ORIGINATING_USER_KEY	O nome de usuário especificado como o ID de usuário de origem no pedido de transferência
SOURCE_AGENT_KEY	O nome do agente que é a origem da transferência
TRANSFER_ID_KEY	O identificador da transferência

Os nomes da chave e os nomes do valor fornecidos na Tabela 2 são constantes definidas na interface TransferMetaDataConstants.

### **Metadados do Arquivo**

Os metadados do arquivo são passados para a saída do início da transferência de origem como parte da especificação do arquivo. Há metadados de arquivos separados para os arquivos de origem e de destino.

Não é possível usar os metadados do arquivo para propagar informações entre diferentes transferências de arquivos.

*Tabela 884. Metadados do Arquivo*

Chave	Valores permitidos	Descrição
CONVERT_LINE_SEPARATORS		O valor da chave usado para que as transferências de texto indiquem se as sequências de separadores de linha CRLF (carriage return-line feed) ou LF (retorno de linha) nos dados de origem devem ser convertidas para a sequência de separadores de linha no destino.
DELIMITER_KEY		O valor da chave usado para definir um delimitador para separar dados de registro ao transferir dados orientados a registro para arquivos normais.  Também usado para transferências de mensagem para arquivo e de arquivo para mensagem.
DELIMITER_POSITION_KEY	DELIMITER_POSITION_PREFIX_VALUE DELIMITER_POSITION_POSTFIX_VALUE	Use com o DELIMITER_KEY para definir a posição do delimitador, prefixo ou sufixo.
DELIMITER_TYPE_KEY	DELIMITER_TYPE_BINARY_VALUE DELIMITER_TYPE_TEXT_VALUE DELIMITER_TYPE_SIZE_VALUE	Use com o DELIMITER_KEY para definir o tipo de delimitador.
DESTINATION_EXIST_KEY	DESTINATION_EXIST_KEY_ERROR_VALUE DESTINATION_EXIST_KEY_OVERWRITE_VALUE	Determina o comportamento da transferência de arquivos se o arquivo de destino existir.
FILE_ALIAS_KEY		O valor da chave usado para definir um alias para o arquivo que está sendo transferido.
FILE_CHECKSUM_METHOD_KEY	FILE_CHECKSUM_METHOD_NONE_VALUE FILE_CHECKSUM_METHOD_MD5_VALUE	Determina o método de soma de verificação a ser usada ao transferir os arquivos.
FILE_CONVERSION_KEY	FILE_CONVERSION_TEXT_VALUE FILE_CONVERSION_BINARY_VALUE	Determina o tipo de conversão aplicado aos conteúdos de arquivos.
FILE_ENCODING_KEY		Determina a codificação usada para um arquivo de texto.
FILE_END_OF_LINE_KEY	FILE_END_OF_LINE_LF_VALUE FILE_END_OF_LINE_CRLF_VALUE	Determina a sequência de caracteres que denota o final de uma linha: <LF > ou <CR> <LF>.
FILE_SPACE_ALIAS		Determina o alias de um arquivo no espaço no arquivo.  <b>Nota:</b> Esses metadados podem ser usados somente se o FILE_TYPE_KEY for FILE_TYPE_FILE_SPACE_VALUE
FILE_SPACE_NAME		Determina o nome do espaço no arquivo.  <b>Nota:</b> Esses metadados podem ser usados somente se o FILE_TYPE_KEY for FILE_TYPE_FILE_SPACE_VALUE

Tabela 884. Metadados do Arquivo (continuação)

Chave	Valores permitidos	Descrição
FILE_TYPE_KEY	FILE_TYPE_FILE_VALUE FILE_TYPE_DIRECTORY_VALUE FILE_TYPE_DATASET_VALUE FILE_TYPE_PDS_VALUE FILE_TYPE_QUEUE_VALUE FILE_TYPE_FILE_SPACE_VALUE	Determina a especificação do arquivo de destino, da fila ou do espaço no arquivo.
GROUP_ID_KEY		O valor da chave usado para que as transferências de mensagem para arquivo determinem o grupo de mensagens a ser lido a partir da fila de origem. Esse atributo apenas é válido quando o valor de USE_GROUPS_KEY é USE_GROUPS_TRUE_VALUE.
INCLUDE_DELIMITER_IN_MESSAGE_KEY	INCLUDE_DELIMITER_IN_MESSAGE_TRUE_VALUE INCLUDE_DELIMITER_IN_MESSAGE_FALSE_VALUE	Valor da chave usado para que as transferências de arquivo para mensagem determinem se devem incluir os delimitadores que foram usados para dividir o arquivo em várias mensagens no final das mensagens. Esse atributo apenas é válido quando o valor de DELIMITER_TYPE_KEY é DELIMITER_TYPE_BINARY_VALUE DELIMITER_TYPE_TEXT_VALUE.
INSERT_RECORD_LINE_SEPARATOR_KEY		O valor da chave usado para que as transferências de arquivos orientados a registro especifiquem se os separadores de linha devem ser inseridos nos dados após cada registro.
KEEP_TRAILING_SPACES_KEY	KEEP_TRAILING_SPACES_TRUE_VALUE KEEP_TRAILING_SPACES_FALSE_VALUE	O valor da chave usado para determinar se os espaços à direita são removidos da leitura de registros de conjuntos de dados de formato de comprimento fixo.
NEW_RECORD_ON_LINE_SEPARATOR_KEY		O valor da chave usado para que as transferências de texto para arquivos orientados a registro especifiquem se os separadores de linha nos dados devem ser incluídos com os dados de registro ou causarem um novo registro (e não serem gravados).
PERSISTENT_KEY	PERSISTENT_TRUE_VALUE PERSISTENT_FALSE_VALUE PERSISTENT_QDEF_VALUE	O valor da chave usado para que as transferências de arquivo para mensagem determinem se as mensagens são persistentes.
SET_MQ_PROPS_KEY	SET_MQ_PROPS_TRUE_VALUE SET_MQ_PROPS_FALSE_VALUE	O valor da chave usado para que as transferências de arquivo para mensagem determinem se as propriedades da mensagem do IBM MQ serão configuradas na primeira mensagem em um arquivo e em quaisquer mensagens gravadas para a fila quando um erro ocorrer.

Tabela 884. Metadados do Arquivo (continuação)

Chave	Valores permitidos	Descrição
UNRECOGNISED_CODE_PAGE_KEY	UNRECOGNISED_CODE_PAGE_FAIL_VALUE UNRECOGNISED_CODE_PAGE_BINARY_VALUE	O valor da chave usado para que as transferências de arquivo para mensagem determinem se uma transferência de modo de texto falhará ou se será executada uma conversão, caso a página de código dos dados não seja reconhecida pelo gerenciador de filas de destino.
USE_GROUPS_KEY	USE_GROUPS_TRUE_VALUE USE_GROUPS_FALSE_VALUE	O valor da chave usado para que as transferências de mensagem para arquivo determinem a transferência apenas de um grupo completo de mensagens de fila de origem.
WAIT_TIME_KEY		Valor da chave usado para que as transferências de mensagem para arquivo determinem o tempo, em segundos, que o agente de origem deve esperar por um dos casos a seguir: <ul style="list-style-type: none"> <li>• Que uma mensagem apareça na fila de origem, se a fila estiver vazia ou tiver ficado vazia, se o valor de USE_GROUPS_KEY for FALSE.</li> <li>• Que um grupo completo apareça na fila de origem, se o valor de USE_GROUPS_KEY for TRUE.</li> </ul>

Os nomes da chave e os nomes do valor fornecidos na Tabela 3 são constantes definidas na interface FileMetaDataConstants.

### **Conceitos relacionados**

[“Interfaces Java para saídas de usuário do MFT” na página 2207](#)

Use os tópicos nesta seção para obter informações de referência sobre as interfaces Java para rotinas de saída do usuário.

### **Tarefas relacionadas**

[Customizando o MFT com saídas de usuário](#)

### **Referências relacionadas**

[“Saídas de usuário do monitor de recurso do MFT” na página 2200](#)

As saídas de usuário de monitor de recurso permitem configurar código customizado a ser executado quando uma condição acionadora de monitor é satisfeita, antes do início da tarefa associada.

[“Propriedades do agente MFT para saídas de usuário” na página 2205](#)

Além das propriedades padrões no arquivo `agent.properties`, há várias propriedades avançadas especificamente para rotinas de saída de usuário. Essas propriedades não estão incluídas assim se você quiser utilizá-las, deve editar manualmente o arquivo `agent.properties`. Se você fizer uma alteração no arquivo `agent.properties` enquanto o agente está em execução, pare e reinicie o agente para selecionar as alterações.

## **Saídas de usuário do monitor de recurso do MFT**

As saídas de usuário de monitor de recurso permitem configurar código customizado a ser executado quando uma condição acionadora de monitor é satisfeita, antes do início da tarefa associada.

Não é recomendado chamar novas transferências diretamente a partir do código de saída de usuário. Em algumas circunstâncias, isso faz com que os arquivos sejam transferidos várias vezes, pois as saídas de usuário não são resilientes a reinicializações de agente.

As saídas de usuário de monitor de recurso usam a infraestrutura existente para saídas de usuário. As saídas de usuário de monitor são chamadas depois que um monitor foi acionado, mas antes que a tarefa correspondente tenha sido executada pela tarefa do monitor. Isto permite que a saída de usuário modifique a tarefa a ser executada e decida se uma tarefa deve continuar ou não. É possível modificar a tarefa de monitor atualizando os metadados do monitor, que são usados para substituição de variável no documento de tarefa criado pela criação do monitor original. Como alternativa, a saída de monitor pode substituir ou atualizar a cadeia XML de definição de tarefa transmitida como parâmetro. A saída do monitor pode retornar um código de resultado de um 'prosseguir' ou 'cancelar' para a tarefa. Se for retornado cancelamento, a tarefa não será iniciada e o monitor não será iniciado novamente até que o recurso monitorado corresponda às condições acionadoras. Se o recurso não tiver sido alterado, o acionador não será iniciado. Como as demais saídas de usuário, é possível encadear as saídas de monitor juntas. Se uma das saídas retornar um código de resultado de cancelamento, o resultado geral será cancelar e a tarefa não será iniciada.

- Um mapa de metadados de ambiente (o mesmo que outras saídas de usuário)
- Um mapa de metadados de monitor incluindo metadados de sistema imutáveis e metadados de usuário mutáveis. Os metadados de sistema imutáveis são os seguintes:
  - FILENAME - nome do arquivo que satisfaz a condição acionadora
  - FILEPATH - caminho para o arquivo que satisfaz a condição acionadora
  - FILESIZE (em bytes - estes metadados podem não estar presentes) - tamanho do arquivo que satisfazia a condição do acionador
  - LASTMODIFIEDDATE (Local) - data em que o arquivo que satisfazia a condição do acionador foi alterada pela última vez. Esta data é expressa como a data local do fuso horário no qual o agente está em execução e é formatada como uma data ISO 8601.
  - LASTMODIFIEDTIME (Local) - hora em formato local em que o arquivo que satisfazia a condição do acionador foi alterado pela última vez. Esta hora é expressa como a hora local do fuso horário em que o agente está em execução e é formatada como um horário ISO 8601.
  - LASTMODIFIEDDATEUTC - data em formato universal em que o arquivo que satisfazia a condição acionadora foi alterado pela última vez. Esta data é expressa como a data local convertida para o fuso horário UTC e é formatada como uma data ISO 8601.
  - LASTMODIFIEDTIMEUTC - hora em formato universal em que o arquivo que satisfazia a condição acionadora foi alterado pela última vez. Esta hora é expressa em hora local convertida para o fuso horário UTC e é formatada como uma hora ISO 8601.
  - AGENTNAME - o nome do agente de monitoramento
- Uma cadeia XML que representa a tarefa a ser executada como resultado do acionador de monitor.

As saídas de monitor retornam os seguintes dados:

- Um indicador que especifica se ainda deve prosseguir (prosseguir ou cancelar)
- Uma cadeia a inserir na mensagem de log satisfeita pelo acionamento

Como resultado da execução do código de saída de monitor, os metadados e a cadeia XML de definição de tarefa do monitor que foram transmitidos como parâmetros poderão também ter sido atualizados.

O valor da propriedade do agente `monitorExitClasses` (no arquivo `agent.properties`) especifica quais classes de saída do monitor serão carregadas, com cada classe de saída separada por uma vírgula. Por exemplo:

```
monitorExitClasses=testExits.TestExit1,testExits.testExit2
```

A interface para a saída de usuário de monitor é:

```
package com.ibm.wmqfte.exitroutine.api;

import java.util.Map;

/**
 * An interface that is implemented by classes that want to be invoked as part of
 * user exit routine processing. This interface defines a method that will be
 * invoked immediately prior to starting a task as the result of a monitor trigger
 */
public interface MonitorExit {

    /**
     * Invoked immediately prior to starting a task as the result of a monitor
     * trigger.
     *
     * @param environmentMetaData
     *      meta data about the environment in which the implementation
     *      of this method is running. This information can only be read,
     *      it cannot be updated by the implementation. The constant
     *      defined in <code>EnvironmentMetaDataConstants</code> class can
     *      be used to access the data held by this map.
     *
     * @param monitorMetaData
     *      meta data to associate with the monitor. The meta data passed
     *      to this method can be altered, and the changes will be
     *      reflected in subsequent exit routine invocations. This map
     *      also contains keys with IBM reserved names. These entries are
     *      defined in the <code>MonitorMetaDataConstants</code> class and
     *      have special semantics. The the values of the IBM reserved names
     *      cannot be modified by the exit
     *
     * @param taskDetails
     *      An XML String representing the task to be executed as a result of
     *      the monitor triggering. This XML string may be modified by the
     *      exit
     *
     * @return
     *      a monitor exit result object which is used to determine if the
     *      task should proceed, or be cancelled.
     */
    MonitorExitResult onMonitor(Map<String, String> environmentMetaData,
                               Map<String, String> monitorMetaData,
                               Reference<String> taskDetails);
}
```

As constantes para os valores reservados pela IBM nos metadados do monitor são as seguintes:

```
package com.ibm.wmqfte.exitroutine.api;

/**
 * Constants for IBM reserved values placed into the monitor meta data
 * maps used by the monitor exit routines.
 */
public interface MonitorMetaDataConstants {

    /**
     * The value associated with this key is the name of the trigger
     * file associated with the monitor. Any modification performed
     * to this property by user exit routines will be ignored.
     */
    final String FILE_NAME_KEY = "FILENAME";

    /**
     * The value associated with this key is the path to the trigger
     * file associated with the monitor. Any modification performed
     * to this property by user exit routines will be ignored.
     */
    final String FILE_PATH_KEY = "FILEPATH";

    /**
     * The value associated with this key is the size of the trigger
     * file associated with the monitor. This will not be present in
     * the cases where the size cannot be determined. Any modification
     * performed to this property by user exit routines will be ignored.
     */
}
```

```

*/
final String FILE_SIZE_KEY = "FILESIZE";

/**
 * The value associated with this key is the local date on which
 * the trigger file associated with the monitor was last modified.
 * Any modification performed to this property by user exit routines
 * will be ignored.
 */
final String LAST_MODIFIED_DATE_KEY = "LASTMODIFIEDDATE";

/**
 * The value associated with this key is the local time at which
 * the trigger file associated with the monitor was last modified.
 * Any modification performed to this property by user exit routines
 * will be ignored.
 */
final String LAST_MODIFIED_TIME_KEY = "LASTMODIFIEDTIME";

/**
 * The value associated with this key is the UTC date on which
 * the trigger file associated with the monitor was last modified.
 * Any modification performed to this property by user exit routines
 * will be ignored.
 */
final String LAST_MODIFIED_DATE_KEY_UTC = "LASTMODIFIEDDATEUTC";

/**
 * The value associated with this key is the UTC time at which
 * the trigger file associated with the monitor was last modified.
 * Any modification performed to this property by user exit routines
 * will be ignored.
 */
final String LAST_MODIFIED_TIME_KEY_UTC = "LASTMODIFIEDTIMEUTC";

/**
 * The value associated with this key is the name of the agent on which
 * the monitor is running. Any modification performed to this property by
 * user exit routines will be ignored.
 */
final String MONITOR_AGENT_KEY = "AGENTNAME";
}

```

## Saída do usuário do monitor de exemplo

Esta classe de exemplo implementa a interface `MonitorExit`. Esse exemplo inclui uma variável de substituição customizada nos metadados do monitor chamados `REDIRECTEDAGENT` que serão preenchidos com um valor de `LONDON` se a hora do dia for ímpar, e um valor de `PARIS` para horas pares. O código de resultado de saída do monitor é configurado para sempre retornar `proceed`.

```

package com.ibm.wmqfte.monitor;

import java.util.Calendar;
import java.util.Map;

import com.ibm.wmqfte.exitroutine.api.MonitorExit;
import com.ibm.wmqfte.exitroutine.api.MonitorExitResult;
import com.ibm.wmqfte.exitroutine.api.Reference;

/**
 * Example resource monitor user exit that changes the monitor mutable
 * metadata value between 'LONDON' and 'PARIS' depending on the hour of the day.
 *
 */
public class TestMonitorExit implements MonitorExit {

    // custom variable that will substitute destination agent
    final static String REDIRECTED_AGENT = "REDIRECTEDAGENT";

    public MonitorExitResult onMonitor(
        Map<String, String> environmentMetaData,
        Map<String, String> monitorMetaData,
        Reference<String> taskDetails) {

        // always succeed
        final MonitorExitResult result = MonitorExitResult.PROCEED_RESULT;

```

```

        final int hour = Calendar.getInstance().get(Calendar.HOUR_OF_DAY);

        if (hour%2 == 1) {
            monitorMetaData.put(REDIRECTED_AGENT, "LONDON");
        } else {
            monitorMetaData.put(REDIRECTED_AGENT, "PARIS");
        }

        return result;
    }
}

```

A tarefa correspondente para um monitor que faz uso da variável de substituição *REDIRECTEDAGENT* poderia assemelhar-se ao seguinte:

```

<?xml version="1.0" encoding="UTF-8"?>
<request version="4.00"
  xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="FileTransfer.xsd">
  <managedTransfer>
    <originator>
      <hostName>reportserver.com</hostName>
      <userID>USER1</userID>
    </originator>
    <sourceAgent agent="AGENT1"
      QMgr="QM1"/>
    <destinationAgent agent="{REDIRECTEDAGENT}"
      QMgr="QM2"/>
    <transferSet>
      <item mode="binary" checksumMethod="MD5">
        <source recursive="false" disposition="delete">
          <file>c:\sourcefiles\reports.doc</file>
        </source>
        <destination type="file" exist="overwrite">
          <file>c:\destinationfiles\reports.doc</file>
        </destination>
      </item>
    </transferSet>
  </managedTransfer>
</request>

```

Antes que essa transferência seja iniciada, o valor do <destinationAgent> atributo agent do elemento é substituído por LONDON ou PARIS.

É necessário especificar a variável de substituição na classe de saída do monitor e a definição de tarefa XML em maiúsculas.

### **Conceitos relacionados**

[Customizando o MFT com saídas de usuário](#)

“Metadados para saídas de usuário do MFT” na página 2196

Há três tipos diferentes de metadados que podem ser fornecidos para rotinas de saída do usuário para o Managed File Transfer: metadados de ambiente, de transferência e de arquivo. Esses metadados são apresentados como mapas de pares de valores e chave Java.

“Interfaces Java para saídas de usuário do MFT” na página 2207

Use os tópicos nesta seção para obter informações de referência sobre as interfaces Java para rotinas de saída do usuário.

[Saídas de usuário de origem e destino do MFT](#)

### **Referências relacionadas**

“Propriedades do agente MFT para saídas de usuário” na página 2205

Além das propriedades padrões no arquivo `agent.properties`, há várias propriedades avançadas especificamente para rotinas de saída de usuário. Essas propriedades não estão incluídas assim se você quiser utilizá-las, deve editar manualmente o arquivo `agent.properties`. Se você fizer uma alteração no arquivo `agent.properties` enquanto o agente está em execução, pare e reinicie o agente para selecionar as alterações.



## Propriedades do agente MFT para saídas de usuário

Além das propriedades padrões no arquivo `agent.properties`, há várias propriedades avançadas especificamente para rotinas de saída de usuário. Essas propriedades não estão incluídas assim se você quiser utilizá-las, deve editar manualmente o arquivo `agent.properties`. Se você fizer uma alteração no arquivo `agent.properties` enquanto o agente está em execução, pare e reinicie o agente para selecionar as alterações.

As variáveis de ambiente podem ser utilizadas em algumas propriedades Managed File Transfer que representam locais de arquivo ou diretório. Isto permite que os locais de arquivos ou diretórios usados ao executar partes do produto variem dependendo das mudanças de ambiente, como que usuário está executando o processo. Para obter mais informações, consulte [Variáveis de ambiente em propriedades do MFT](#)

## Propriedades da rotina de saída de usuário

As rotinas de saída de usuário são chamadas na ordem listadas na tabela a seguir: Para obter mais informações sobre o arquivo `agent.properties`, consulte [Propriedades avançadas do agente: rotina de saída do usuário](#).


Nome da Propriedade	Descrição
<code>sourceTransferEndExitClasses</code>	Especifica uma lista separada por vírgula de classes que implementam a rotina de saída de encerramento de uma transferência de origem.
<code>sourceTransferStartExitClasses</code>	Especifica uma lista separada por vírgula de classes que implementam a rotina de saída de início de uma transferência de origem.
<code>destinationTransferStartExitClasses</code>	Especifica uma lista separada por vírgula de classes que implementam a rotina de saída de usuário no início de uma transferência de destino.
<code>destinationTransferEndExitClasses</code>	Especifica uma lista separada por vírgula de classes que implementam uma rotina de saída do usuário de transferência de destino.
<code>exitClassPath</code>	<p>Define uma lista de diretórios, específica da plataforma e delimitada por caracteres, que atua como o caminho de classe das rotinas de saída de usuário.</p> <p>O diretório de saídas do agente é procurado antes de quaisquer entradas nesse caminho de classe</p> <p>Parênteses, vírgulas (,) e barras invertidas (\) são caracteres especiais em comandos MFT e devem ser escapados com um caractere de barra invertida (\).</p> <p> Caminhos de arquivo no Windows podem ser especificados usando barras duplas invertidas (\\) como separador ou usando barras simples (/). Por exemplo:</p> <pre>exitClassPath=C:/mycomp/mqft/exits/encryptFileExit.jar; C:/mycomp/mqft/exits/fileFilter.jar.</pre> <p>O valor dessa propriedade pode conter variáveis de ambiente</p>
<code>exitNativeLibraryPath</code>	<p>Define uma lista de diretórios, específica da plataforma e delimitada por caracteres, que atua como o caminho de biblioteca nativa das rotinas de saída de usuário.</p> <p>O valor dessa propriedade pode conter variáveis de ambiente</p>
<code>monitorExitClasses</code>	Especifica uma lista separada por vírgula de classes que implementam uma rotina de saída do monitor. Para mais informações, consulte <a href="#">MFT saídas de usuário do monitor de recursos</a> .
<code>protocolBridgeCredentialExitClasses</code>	Especifica uma lista de classes separadas por vírgula que implementam uma rotina de saída de usuário de credenciais de ponte de protocolo. Para obter mais informações, consulte <a href="#">Mapeando credenciais para um servidor de arquivos usando classes de saída</a> .

Tabela 885. Propriedades do Agente para Saídas de Usuário (continuação)

Nome da Propriedade	Descrição
protocolBridgePropertiesExitClasses	Especifica uma lista separada por vírgula de classes que implementam uma rotina de saída do usuário de propriedades de servidor de ponte de protocolo. Para obter mais informações, consulte <a href="#">ProtocolBridgePropertiesExit2: consultando propriedades do servidor de arquivos de protocolo</a> .
IOExitClasses	Especifica uma lista separada por vírgula de classes que implementam uma rotina de saída do usuário de E/S. Liste apenas as classes que implementam a interface IOExit, ou seja, não liste classes que implementam as outras interfaces de saída de usuário de E/S, por exemplo, IOExitResourcePath e IOExitChannel. Para mais informações, consulte <a href="#">Usando MFT saídas de usuário de E/S de transferência</a> .

## Ordem de Chamada de Saída

As saídas de origem e de destino são chamadas na seguinte ordem:

1. SourceTransferStartExit
2. DestinationTransferStartExit
3. DestinationTransferEndExit
4. SourceTransferEndExit

## Encadeando Saídas de Origem e de Destino

Se você especificar várias saídas, a primeira saída na lista será chamada primeiro, seguida pela segunda saída e assim por diante. Quaisquer alterações feitas pela primeira saída serão passadas como entrada para a saída subsequente e assim sucessivamente. Por exemplo, se houver duas saídas de início de transferência de origem, quaisquer alterações feitas nos metadados de transferência pela primeira saída serão entrada da segunda. Cada saída retorna seu próprio resultado. Se todas as saídas de determinado tipo retornarem PROCEED como código de resultado da transferência, o resultado geral será PROCEED. Se uma ou mais saída retornar CANCEL\_TRANSFER, o resultado geral será CANCEL\_TRANSFER. Todo os códigos de resultado e cadeias retornados pelas saídas são registrados no log de transferência.

Se o resultado geral da saída de início de transferência de origem for PROCEED, a transferência prosseguirá usando quaisquer alterações feitas pelas saídas. Se o resultado geral for CANCEL\_TRANSFER, as saídas do término da transferência de origem serão chamadas e a transferência, cancelada. O status de conclusão no log de transferência será "cancelado".

Se o resultado geral das saídas de início de transferência de destino for PROCEED, a transferência prosseguirá usando quaisquer alterações feitas pelas saídas. Se o resultado geral for CANCEL\_TRANSFER, as saídas do término da transferência de destino serão invocadas e, depois, as saídas do término da transferência de origem serão invocadas. Por último, a transferência será cancelada. O status de conclusão no log de transferência será "cancelado".

Se uma saída de origem ou de destino precisar transmitir informações para as seguintes saídas na cadeia ou na ordem de execução, isso deve ser feito atualizando os metadados de transferência. O uso dos metadados de transferência é específico da implementação da saída. Por exemplo, se uma saída configurar o resultado de retorno como CANCEL\_TRANSFER e precisar se comunicar com as seguintes saídas canceladas pela transferência, isso deve ser feito configurando um valor de metadados de transferência de uma maneira entendida pelas outras saídas.

### exemplo

```
sourceTransferStartExitClasses=com.ibm.wmqfte.test.MFTTestSourceTransferStartExit
sourceTransferEndExitClasses=com.ibm.wmqfte.test.MFTTestSourceTransferEndExit
destinationTransferStartExitClasses=com.ibm.wmqfte.test.MFTTestDestinationTransferStartExit
destinationTransferEndExitClasses=com.ibm.wmqfte.test.MFTTestDestinationTransferEndExit
exitClassPath=C:/mycomp/mqft/exits/encryptFileExit.jar;C:/mycomp/mqft/exits/fileFilter.jar
```

## **Conceitos relacionados**

[Customizando o MFT com saídas de usuário](#)

[“Metadados para saídas de usuário do MFT” na página 2196](#)

Há três tipos diferentes de metadados que podem ser fornecidos para rotinas de saída do usuário para o Managed File Transfer: metadados de ambiente, de transferência e de arquivo. Esses metadados são apresentados como mapas de pares de valores e chave Java.

[“Interfaces Java para saídas de usuário do MFT” na página 2207](#)

Use os tópicos nesta seção para obter informações de referência sobre as interfaces Java para rotinas de saída do usuário.

## **Referências relacionadas**

[“Saídas de usuário do monitor de recurso do MFT” na página 2200](#)

As saídas de usuário de monitor de recurso permitem configurar código customizado a ser executado quando uma condição acionadora de monitor é satisfeita, antes do início da tarefa associada.

[Variáveis de ambiente em propriedades MFT](#)

[O arquivo MFT `agent.properties`](#)

## **Interfaces Java para saídas de usuário do MFT**

Use os tópicos nesta seção para obter informações de referência sobre as interfaces Java para rotinas de saída do usuário.

### **Tarefas relacionadas**

[Customizando o MFT com saídas de usuário](#)

### **Referências relacionadas**

[“Interface `DestinationTransferStartExit.java`” na página 2211](#)

[“Interface `DestinationTransferEndExit.java`” na página 2209](#)

[“Interface `IOExit.java`” na página 2213](#)

[“Interface `IOExitChannel.java`” na página 2215](#)

[“Interface `IOExitLock.java`” na página 2217](#)

[“Interface `IOExitPath.java`” na página 2217](#)

[“Interface `IOExitProperties.java`” na página 2218](#)

[“Interface `IOExitRecordChannel.java`” na página 2221](#)

[“Interface `IOExitRecordResourcePath.java`” na página 2223](#)

[“Interface `IOExitResourcePath.java`” na página 2224](#)

[“Interface `IOExitWildcardPath.java`” na página 2229](#)

[“Interface `MonitorExit.java`” na página 2229](#)

[“Interface `ProtocolBridgeCredentialExit.java`” na página 2230](#)

[“Interface `ProtocolBridgeCredentialExit2.java`” na página 2231](#)

[“Interface `ProtocolBridgePropertiesExit2.java`” na página 2232](#)

[“Interface `SourceTransferStartExit.java`” na página 2235](#)

[“Interface `SourceTransferEndExit.java`” na página 2234](#)

## ***Interface `CDCredentialExit.java`***

### **`CDCredentialExit.java`**

```
/*  
 * Licensed Materials - Property of IBM  
 *  
 * "Restricted Materials of IBM"
```

```

*
* 5724-H72
*
*   Copyright IBM Corp. 2011, 2024. All Rights Reserved.
*
* US Government Users Restricted Rights - Use, duplication or
* disclosure restricted by GSA ADP Schedule Contract with
* IBM Corp.
*/
package com.ibm.wmqfte.exitroutine.api;

import java.util.Map;

/**
 * An interface that is implemented by classes that are invoked as part of
 * user exit routine processing. This interface defines methods that are
 * invoked by a Connect:Direct bridge agent to map the IBM MQ user ID of the transfer to credentials
 * that are used to access the Connect:Direct node.
 * There will be one instance of each implementation class per Connect:Direct bridge agent. The methods
 * can be called from different threads so the methods must be synchronized.
 */
public interface CDCredentialExit {

    /**
     * Invoked once when a Connect:Direct bridge agent is started. It is intended to initialize
     * any resources that are required by the exit
     *
     * @param bridgeProperties
     *         The values of properties defined for the Connect:Direct bridge.
     *         These values can only be read, they cannot be updated by
     *         the implementation.
     *
     * @return true if the initialisation is successful and false if unsuccessful
     *         If false is returned from an exit the Connect:Direct bridge agent does not
     *         start.
     */
    public boolean initialize(final Map<String, String> bridgeProperties);

    /**
     * Invoked once per transfer to map the IBM MQ user ID in the transfer message to the
     * credentials to be used to access the Connect:Direct node.
     *
     * @param mqUserId The IBM MQ user ID from which to map to the credentials to be used
     *                 to access the Connect:Direct node
     * @param snode    The name of the Connect:Direct SNODE specified as the cdNode in the
     *                 file path. This is used to map the correct user ID and password for the
     *                 SNODE.
     * @return        A credential exit result object that contains the result of the map and
     *                 the credentials to use to access the Connect:Direct node
     */
    public CDCredentialExitResult mapMQUserId(final String mqUserId, final String snode);

    /**
     * Invoked once when a Connect:Direct bridge agent is shutdown. This method releases
     * any resources that were allocated by the exit
     *
     * @param bridgeProperties
     *         The values of properties defined for the Connect:Direct bridge.
     *         These values can only be read, they cannot be updated by
     *         the implementation.
     *
     * @return
     */
    public void shutdown(final Map<String, String> bridgeProperties);
}

```

## ***Interface CredentialExitResult.java***

### **CredentialExitResult.java**

```

/*
 * Licensed Materials - Property of IBM
 *
 * "Restricted Materials of IBM"
 *
 * 5724-H72
 *
 *   Copyright IBM Corp. 2008, 2024. All Rights Reserved.

```

```

*
* US Government Users Restricted Rights - Use, duplication or
* disclosure restricted by GSA ADP Schedule Contract with
* IBM Corp.
*/

package com.ibm.wmqfte.exitroutine.api;

/**
 * The result of invoking a Credential mapMQUserId exit method. It is composed of a result
 * code, which determines whether the mapping of the user id was successful, and an optional
 * Credentials object if the mapping is successful.
 */
public class CredentialExitResult {

    private final CredentialExitResultCode resultCode;
    private final Credentials credentials;

    /**
     * Constructor. Creates a credential exit result object with a specified result
     * code and optionally credentials.
     *
     * @param resultCode
     *     The result code to associate with the exit result being created.
     *
     * @param credentials
     *     The credentials to associate with the exit result being created.
     *     A value of <code>null</code> can be specified to indicate no
     *     credentials. If the resultCode is USER_SUCCESSFULLY_MAPPED the
     *     credentials must be set to a non-null value,
     */
    public CredentialExitResult(CredentialExitResultCode resultCode, Credentials credentials) {
        this.resultCode = resultCode;
        this.credentials = credentials;
    }

    /**
     * Returns the result code associated with this credential exit result
     *
     * @return
     *     the result code associated with this exit result.
     */
    public CredentialExitResultCode getResultCode() {
        return resultCode;
    }

    /**
     * Returns the credentials associated with this credential exit result
     *
     * @return
     *     the explanation associated with this credential exit result.
     */
    public Credentials getCredentials() {
        return credentials;
    }
}

```

## Tarefas relacionadas

[Customizando o MFT com saídas de usuário](#)

## Referências relacionadas

[“Interface SourceTransferStartExit.java” na página 2235](#)

[“Interface DestinationTransferStartExit.java” na página 2211](#)

[“Interface DestinationTransferEndExit.java” na página 2209](#)

[“Interface MonitorExit.java” na página 2229](#)

[“Interface ProtocolBridgeCredentialExit.java” na página 2230](#)

## ***Interface DestinationTransferEndExit.java***

### **DestinationTransferEndExit.java**

```

/*
 * Licensed Materials - Property of IBM
 *
 */

```

```

* "Restricted Materials of IBM"
*
* 5724-H72
*
* □ Copyright IBM Corp. 2008, 2024. All Rights Reserved.
*
* US Government Users Restricted Rights - Use, duplication or
* disclosure restricted by GSA ADP Schedule Contract with
* IBM Corp.
*/
package com.ibm.wmqfte.exitpoint.api;

/**
 * An interface that is implemented by classes that want to be invoked as part of
 * user exit routine processing. This interface defines a method that will be
 * invoked immediately after completing a transfer on the agent acting as the
 * destination of the transfer.
 */
public interface DestinationTransferEndExit {

    /**
     * Invoked immediately after the completion of a transfer on the agent acting as
     * the destination of the transfer.
     *
     * @param transferExitResult
     *     a result object reflecting whether or not the transfer completed
     *     successfully.
     *
     * @param sourceAgentName
     *     the name of the agent acting as the source of the transfer.
     *
     * @param destinationAgentName
     *     the name of the agent acting as the destination of the
     *     transfer. This is the name of the agent that the
     *     implementation of this method will be invoked from.
     *
     * @param environmentMetaData
     *     meta data about the environment in which the implementation
     *     of this method is running. This information can only be read,
     *     it cannot be updated by the implementation. The constants
     *     defined in <code>EnvironmentMetaDataConstants</code> class can
     *     be used to access the data held by this map.
     *
     * @param transferMetaData
     *     meta data to associate with the transfer. The information can
     *     only be read, it cannot be updated by the implementation. This
     *     map may also contain keys with IBM reserved names. These
     *     entries are defined in the <code>TransferMetaDataConstants</code>
     *     class and have special semantics.
     *
     * @param fileResults
     *     a list of file transfer result objects that describe the source
     *     file name, destination file name and result of each file transfer
     *     operation attempted.
     *
     * @return
     *     an optional description to enter into the log message describing
     *     transfer completion. A value of <code>null</code> can be used
     *     when no description is required.
     */
    String onDestinationTransferEnd(TransferExitResult transferExitResult,
        String sourceAgentName,
        String destinationAgentName,
        Map<String, String>environmentMetaData,
        Map<String, String>transferMetaData,
        List<FileTransferResult>fileResults);
}

```

## Tarefas relacionadas

[Customizando o MFT com saídas de usuário](#)

## Referências relacionadas

[“Interface SourceTransferStartExit.java” na página 2235](#)

[“Interface SourceTransferEndExit.java” na página 2234](#)

[“Interface DestinationTransferStartExit.java” na página 2211](#)

[“Interface MonitorExit.java” na página 2229](#)

[“Interface ProtocolBridgeCredentialExit.java” na página 2230](#)

## Interface DestinationTransferStartExit.java

### DestinationTransferStartExit.java

```
/*
 * Licensed Materials - Property of IBM
 *
 * "Restricted Materials of IBM"
 *
 * 5724-H72
 *
 * © Copyright IBM Corp. 2008, 2024. All Rights Reserved.
 *
 * US Government Users Restricted Rights - Use, duplication or
 * disclosure restricted by GSA ADP Schedule Contract with
 * IBM Corp.
 */
package com.ibm.wmqfte.exitpoint.api;

/**
 * An interface that is implemented by classes that want to be invoked as part of
 * user exit routine processing. This interface defines a method that will be
 * invoked immediately prior to starting a transfer on the agent acting as the
 * destination of the transfer.
 */
public interface DestinationTransferStartExit {

    /**
     * Invoked immediately prior to starting a transfer on the agent acting as
     * the destination of the transfer.
     *
     * @param sourceAgentName
     *     the name of the agent acting as the source of the transfer.
     *
     * @param destinationAgentName
     *     the name of the agent acting as the destination of the
     *     transfer. This is the name of the agent that the
     *     implementation of this method will be invoked from.
     *
     * @param environmentMetaData
     *     meta data about the environment in which the implementation
     *     of this method is running. This information can only be read,
     *     it cannot be updated by the implementation. The constants
     *     defined in EnvironmentMetaDataConstants class can
     *     be used to access the data held by this map.
     *
     * @param transferMetaData
     *     meta data to associate with the transfer. The information can
     *     only be read, it cannot be updated by the implementation. This
     *     map may also contain keys with IBM reserved names. These
     *     entries are defined in the TransferMetaDataConstants
     *     class and have special semantics.
     *
     * @param fileSpecs
     *     a list of file specifications that govern the file data to
     *     transfer. The implementation of this method can modify the
     *     entries in this list and the changes will be reflected in the
     *     files transferred. However, new entries may not be added and
     *     existing entries may not be removed.
     *
     * @return
     *     a transfer exit result object which is used to determine if the
     *     transfer should proceed, or be cancelled.
     */
    TransferExitResult onDestinationTransferStart(String sourceAgentName,
        String destinationAgentName,
        Map<String, String> environmentMetaData,
        Map<String, String> transferMetaData,
        List<Reference<String>> fileSpecs);
}
```

### Tarefas relacionadas

[Customizando o MFT com saídas de usuário](#)

### Referências relacionadas

[“Interface SourceTransferStartExit.java” na página 2235](#)

[“Interface SourceTransferEndExit.java” na página 2234](#)

[“Interface DestinationTransferEndExit.java” na página 2209](#)

[“Interface MonitorExit.java” na página 2229](#)

[“Interface ProtocolBridgeCredentialExit.java” na página 2230](#)

## ***Interface FileTransferResult.java***

### **FileTransferResult.java**

```
/*
 * Licensed Materials - Property of IBM
 *
 * "Restricted Materials of IBM"
 *
 * 5724-H72
 *
 * © Copyright IBM Corp. 2008, 2024. All Rights Reserved.
 *
 * US Government Users Restricted Rights - Use, duplication or
 * disclosure restricted by GSA ADP Schedule Contract with
 * IBM Corp.
 */

package com.ibm.wmqfte.exitroutine.api;

/**
 * Result information about a file transfer.
 */
public interface FileTransferResult {

    /** An enumeration for the <code>getCorrelatorType()</code> method. */
    public enum CorrelationInformationType {
        /** No correlation information is available for this result */
        NONE,
        /**
         * The correlation information relates to work done in
         * IBM Sterling File Gateway.
         */
        SFG
    }

    /**
     * Returns the source file specification, from which the file was transferred.
     *
     * @return the source file specification, from which the file was
     * transferred.
     */
    String getSourceFileSpecification();

    /**
     * Returns the destination file specification, to which the file was transferred.
     *
     * @return the destination file specification, to which the file was
     * transferred. A value of <code>null</code> may be returned
     * if the transfer did not complete successfully.
     */
    String getDestinationFileSpecification();

    /**
     * Returns the result of the file transfer operation.
     *
     * @return the result of the file transfer operation.
     */
    FileExitResult getExitResult();

    /**
     * @return an enumerated value that identifies the product to which this correlating
     * information relates.
     */
    CorrelationInformationType getCorrelatorType();

    /**
     * @return the first string component of the correlating identifier that relates
     * this transfer result to work done in another product. A value of null
     * may be returned either because the other product does not utilize a
     * string based correlation information or because there is no correlation
     * information.
     */
}
```



```

    */
    String getString1Correlator();

    /**
     * @return the first long component of the correlating identifier that relates
     *         this transfer result to work done in another product. A value of zero
     *         is returned when there is no correlation information or the other
     *         product does not utilize long based correlation information or because
     *         the value really is zero!
     */
    long getLong1Correlator();
}

```

## Tarefas relacionadas

[Customizando o MFT com saídas de usuário](#)

## Referências relacionadas

[“Interface SourceTransferStartExit.java” na página 2235](#)

[“Interface DestinationTransferStartExit.java” na página 2211](#)

[“Interface DestinationTransferEndExit.java” na página 2209](#)

[“Interface MonitorExit.java” na página 2229](#)

[“Interface ProtocolBridgeCredentialExit.java” na página 2230](#)

## Interface IOExit.java

### IOExit.java

```

/*
 * Licensed Materials - Property of IBM
 *
 * "Restricted Materials of IBM"
 *
 * 5724-H72
 *
 * © Copyright IBM Corp. 2011, 2024. All Rights Reserved.
 *
 * US Government Users Restricted Rights - Use, duplication or
 * disclosure restricted by GSA ADP Schedule Contract with
 * IBM Corp.
 */
package com.ibm.wmqfte.exitroutine.api;

import java.io.IOException;
import java.util.Map;

import com.ibm.wmqfte.exitroutine.api.IOExitRecordResourcePath.RecordFormat;

/**
 * An interface that is implemented by classes that you want to be invoked as
 * part of user exit routine processing. This interface defines methods that
 * will be invoked during transfers to perform the underlying file system I/O
 * work for WMQFTE transfers.
 * <p>
 * The {@link #initialize(Map)} method will be called once when the exit is
 * first installed. The WMQFTE agent properties are passed to this method, thus
 * enabling the exit to understand its environment.
 * <p>
 * The {@link #isSupported(String)} method will be invoked during WMQFTE
 * transfers to determine whether the user exit should be used. If the
 * {@link #isSupported(String)} method returns a value of {@code true}, the
 * {@link #newPath(String)} method will be invoked for the paths specified for
 * the transfer request. The returned {@link IOExitPath} instance from a
 * {@link #newPath(String)} method invocation will then be used by the WMQFTE
 * transfer to obtain information about the resource and to transfer data to or
 * from the resource.
 * <p>
 * To obtain transfer context for an I/O exit, a {@link SourceTransferStartExit}
 * or {@link DestinationTransferStartExit} as appropriate, should be installed
 * to enable information to be seen by this exit. The
 * {@link SourceTransferStartExit} or {@link DestinationTransferStartExit} are
 * passed the transfer's environment, metadata, and a list of file
 * specifications for the transfer. The paths for the file specifications are
 * the paths passed to the I/O exit's {@link #newPath(String)} method.

```

```

* <p>
* Note also that the {@link #isSupported(String)} and {@link #newPath(String)}
* methods might be called at other times by a WMQFTE agent and not just during
* transfers. For example, at transfer setup time the I/O system is queried to
* resolve the full resource paths for transfer.
*/
public interface IOExit {

    /**
     * Invoked once when the I/O exit is first required for use. It is intended
     * to initialize any resources that are required by the exit.
     *
     * @param agentProperties
     *     The values of properties defined for the WMQFTE agent. These
     *     values can only be read, they cannot be updated by the
     *     implementation.
     * @return {@code true} if the initialization is successful and {@code
     *     false} if unsuccessful. If {@code false} is returned from an
     *     exit, the exit will not be used.
     */
    boolean initialize(final Map<String, String> agentProperties);

    /**
     * Indicates whether this I/O user exit supports the specified path.
     * <p>
     * This method is used by WMQFTE to determine whether the I/O user exit
     * should be used within a transfer. If no I/O user exit returns true for
     * this method, the default WMQFTE file I/O function will be used.
     *
     * @param path
     *     The path to the required I/O resource.
     * @return {@code true} if the specified path is supported by the I/O exit,
     *     {@code false} otherwise
     */
    boolean isSupported(String path);

    /**
     * Obtains a new {@link IOExitPath} instance for the specified I/O resource
     * path.
     * <p>
     * This method will be invoked by WMQFTE only if the
     * {@link #isSupported(String)} method has been called for the path and
     * returned {@code true}.
     *
     * @param path
     *     The path to the required I/O resource.
     * @return A {@link IOExitPath} instance for the specified path.
     * @throws IOException
     *     If the path cannot be created for any reason.
     */
    IOExitPath newPath(String path) throws IOException;

    /**
     * Obtains a new {@link IOExitPath} instance for the specified I/O resource
     * path and passes record format and length information required by the
     * WMQFTE transfer.
     * <p>
     * Typically this method will be called for the following cases:
     * <ul>
     * <li>A path where a call to {@link #newPath(String)} has previously
     * returned a {@link IOExitRecordResourcePath} instance and WMQFTE is
     * re-establishing a new {@link IOExitPath} instance for the path, from an
     * internally-serialized state. The passed recordFormat and recordLength
     * will be the same as those for the original
     * {@link IOExitRecordResourcePath} instance.</li>
     * <li>A transfer destination path where the source of the transfer is
     * record oriented. The passed recordFormat and recordLength will be the
     * same as those for the source.</li>
     * </ul>
     * The implementation can act on the record format and length information as
     * deemed appropriate. For example, for a destination agent if the
     * destination does not already exist and the source of the transfer is
     * record oriented, the passed recordFormat and recordLength information
     * could be used to create an appropriate record-oriented destination path.
     * If the destination path already exists, the passed recordFormat and
     * recordLength information could be used to perform a compatibility check
     * and throw an {@link IOException} if the path is not compatible. A
     * compatibility check could ensure that a record oriented path's record
     * format is the same as the passed record format or that the record length
     * is greater or equal to the passed record length.
     * <p>
     * This method will be invoked by WMQFTE only if the

```

```

* {@link #isSupported(String)} method has been called for the path and
* returned {@code true}.
*
* @param path
*     The path to the required I/O resource.
* @param recordFormat
*     The advised record format.
* @param recordLength
*     The advised record length.
* @return A {@link IOExitPath} instance for the specified path.
* @throws IOException
*     If the path cannot be created for any reason. For example,
*     the passed record format or length is incompatible with the
*     path's actual record format or length.
*/
IOExitPath newPath(String path, RecordFormat recordFormat, int recordLength)
    throws IOException;

```

## Tarefas relacionadas

[Usando as saídas de usuário de E/S de transferência do MFT](#)

[Customizando o MFT com saídas de usuário](#)

## Interface *IOExitChannel.java*

### IOExitChannel.java

```

/*
 * Licensed Materials - Property of IBM
 *
 * "Restricted Materials of IBM"
 *
 * 5724-H72
 *
 * © Copyright IBM Corp. 2011, 2024. All Rights Reserved.
 *
 * US Government Users Restricted Rights - Use, duplication or
 * disclosure restricted by GSA ADP Schedule Contract with
 * IBM Corp.
 */
package com.ibm.wmqfte.exitroutine.api;

import java.io.IOException;
import java.nio.ByteBuffer;

/**
 * Represents a channel that enables data to be read from or written to an
 * {@link IOExitResourcePath} resource.
 */
public interface IOExitChannel {

    /**
     * Obtains the data size for the associated {@link IOExitResourcePath} in
     * bytes.
     *
     * @return The data size in bytes.
     * @throws IOException
     *     If a problem occurs while attempting obtain the size.
     */
    long size() throws IOException;

    /**
     * Closes the channel, flushing any buffered write data to the resource and
     * releasing any locks.
     *
     * @throws RecoverableIOException
     *     If a recoverable problem occurs while closing the resource.
     *     This means that WMQFTE can attempt to recover the transfer.
     * @throws IOException
     *     If some other I/O problem occurs. For example, the channel might
     *     already be closed.
     */
    void close() throws RecoverableIOException, IOException;

    /**
     * Reads data from this channel into the given buffer, starting at this
     * channel's current position, and updates the current position by the

```

```

* amount of data read.
* <p>
* Data is copied into the buffer starting at its current position and up to
* its limit. On return, the buffer's position is updated to reflect the
* number of bytes read.
*
* @param buffer
*       The buffer that the data is to be copied into.
* @return The number of bytes read, which might be zero, or -1 if the end of
*         data has been reached.
* @throws RecoverableIOException
*         If a recoverable problem occurs while reading the data. For a
*         WMQFTE transfer this means that it will attempt to recover.
* @throws IOException
*         If some other I/O problem occurs. For a WMQFTE transfer this
*         means that it will be failed.
*/
int read(ByteBuffer buffer) throws RecoverableIOException, IOException;

/**
* Writes data to this channel from the given buffer, starting at this
* channel's current position, and updates the current position by the
* amount of data written. The channel's resource is grown to accommodate
* the data, if necessary.
* <p>
* Data is copied from the buffer starting at its current position and up to
* its limit. On return, the buffer's position is updated to reflect the
* number of bytes written.
*
* @param buffer
*       The buffer containing the data to be written.
* @return The number of bytes written, which might be zero.
* @throws RecoverableIOException
*         If a recoverable problem occurs while writing the data. For a
*         WMQFTE transfer this means that it will attempt to recover.
* @throws IOException
*         If some other I/O problem occurs. For a WMQFTE transfer this
*         means that it will be failed.
*/
int write(ByteBuffer buffer) throws RecoverableIOException, IOException;

/**
* Forces any updates to this channel's resource to be written to its
* storage device.
* <p>
* This method is required to force changes to both the resource's content
* and any associated metadata to be written to storage.
*
* @throws RecoverableIOException
*         If a recoverable problem occurs while performing the force.
*         For a WMQFTE transfer this means that it will attempt to
*         recover.
* @throws IOException
*         If some other I/O problem occurs. For a WMQFTE transfer this
*         means that it will be failed.
*/
void force() throws RecoverableIOException, IOException;

/**
* Attempts to lock the entire resource associated with the channel for
* shared or exclusive access.
* <p>
* The intention is for this method not to block if the lock is currently
* unavailable.
*
* @param shared
*       {@code true} if a shared lock is required, {@code false} if an
*       exclusive lock is required.
* @return A {@link IOExitLock} instance representing the newly acquired
*         lock or null if the lock cannot be obtained.
* @throws IOException
*         If a problem occurs while attempting to acquire the lock.
*/
IOExitLock tryLock(boolean shared) throws IOException;
}

```

## Tarefas relacionadas

[Usando as saídas de usuário de E/S de transferência do MFT](#)

[Customizando o MFT com saídas de usuário](#)

## Interface IOExitLock.java

### IOExitLock.java

```
/*
 * Licensed Materials - Property of IBM
 *
 * "Restricted Materials of IBM"
 *
 * 5724-H72
 *
 * © Copyright IBM Corp. 2011, 2024. All Rights Reserved.
 *
 * US Government Users Restricted Rights - Use, duplication or
 * disclosure restricted by GSA ADP Schedule Contract with
 * IBM Corp.
 */
package com.ibm.wmqfte.exitroutine.api;

import java.io.IOException;

/**
 * Represents a lock on a resource for either shared or exclusive access.
 * {@link IOExitLock} instances are returned from
 * {@link IOExitChannel#tryLock(boolean)} calls and WMQFTE will request the
 * release of the lock at the appropriate time during a transfer. Additionally, when
 * a {@link IOExitChannel#close()} method is called it will be the
 * responsibility of the channel to release any associated locks.
 */
public interface IOExitLock {

    /**
     * Releases the lock.
     * <p>
     * After this method has been successfully called the lock is to be deemed as invalid.
     *
     * @throws IOException
     *         If the channel associated with the lock is not open or
     *         another problem occurs while attempting to release the lock.
     */
    void release() throws IOException;

    /**
     * Indicates whether this lock is valid.
     * <p>
     * A lock is considered valid until its @ {@link #release()} method is
     * called or the associated {@link IOExitChannel} is closed.
     *
     * @return {@code true} if this lock is valid, {@code false} otherwise.
     */
    boolean isValid();

    /**
     * @return {@code true} if this lock is for shared access, {@code false} if
     *         this lock is for exclusive access.
     */
    boolean isShared();
}
```

### Tarefas relacionadas

[Usando as saídas de usuário de E/S de transferência do MFT](#)  
[Customizando o MFT com saídas de usuário](#)

## Interface IOExitPath.java

### IOExitPath.java

```
/*
 * Licensed Materials - Property of IBM
 *
 * "Restricted Materials of IBM"
 *
 * 5724-H72
```

```

*
*   Copyright IBM Corp. 2011, 2024. All Rights Reserved.
*
*   US Government Users Restricted Rights - Use, duplication or
*   disclosure restricted by GSA ADP Schedule Contract with
*   IBM Corp.
*/
package com.ibm.wmqfte.exitroutine.api;

/**
 * Represents an abstract path that can be inspected and queried by WMQFTE for
 * transfer purposes.
 * <p>
 * There are two types of path supported:
 * <ul>
 * <li>{@link IOExitResourcePath} - Represents a path that denotes a data
 * resource. For example, a file, directory, or group of database records.</li>
 * <li>{@link IOExitWildcardPath} - Represents a wildcard path that can be
 * expanded to multiple {@link IOExitResourcePath} instances.</li>
 * </ul>
 */
public abstract interface IOExitPath {

    /**
     * Obtains the abstract path as a {@link String}.
     *
     * @return The abstract path as a {@link String}.
     */
    String getPath();

    /**
     * Obtains the name portion of this abstract path as a {@link String}.
     * <p>
     * For example, a UNIX-style file system implementation evaluates the
     * path {@code /home/fteuser/file1.txt} as having a name of {@code
     * file1.txt}.
     *
     * @return the name portion of this abstract path as a {@link String}.
     */
    String getName();

    /**
     * Obtains the parent path for this abstract path as a {@link String}.
     * <p>
     * For example, a UNIX-style file system implementation evaluates the
     * path {@code /home/fteuser/file1.txt} as having a parent path of {@code
     * /home/fteuser}.
     *
     * @return The parent portion of the path as a {@link String}.
     */
    String getParent();

    /**
     * Obtains the abstract paths that match this abstract path.
     * <p>
     * If this abstract path denotes a directory resource, a list of paths
     * for all resources within the directory are returned.
     * <p>
     * If this abstract path denotes a wildcard, a list of all paths
     * matching the wildcard are returned.
     * <p>
     * Otherwise null is returned, because this abstract path probably denotes a
     * single file resource.
     *
     * @return An array of {@link IOExitResourcePath}s that
     *         match this path, or null if this method is not applicable.
     */
    IOExitResourcePath[] listPaths();
}

```

### Tarefas relacionadas

[Usando as saídas de usuário de E/S de transferência do MFT](#)  
[Customizando o MFT com saídas de usuário](#)

### ***Interface IOExitProperties.java***

## IOExitProperties.java

```
/*
 * Licensed Materials - Property of IBM
 *
 * "Restricted Materials of IBM"
 *
 * 5724-H72
 *
 * © Copyright IBM Corp. 2011, 2024. All Rights Reserved.
 *
 * US Government Users Restricted Rights - Use, duplication or
 * disclosure restricted by GSA ADP Schedule Contract with
 * IBM Corp.
 */
package com.ibm.wmqfte.exitroutine.api;

/**
 * Properties that determine how WMQFTE treats an {@link IOExitPath} for certain
 * aspects of I/O. For example, whether to use intermediate files.
 */
public class IOExitProperties {

    private boolean rereadSourceOnRestart = true;
    private boolean rechecksumSourceOnRestart = true;
    private boolean rechecksumDestinationOnRestart = true;
    private boolean useIntermediateFileAtDestination = true;
    private boolean requiresSingleThreadedChannelIO = false;

    /**
     * Determines whether the I/O exit implementation expects the resource to be
     * re-read from the start if a transfer is restarted.
     *
     * @return {@code true} if, on restart, the I/O exit expects the source
     * resource to be opened at the beginning and re-read from the
     * beginning (the {@link IOExitPath#openForRead(long)} method is
     * always invoked with 0L as an argument). {@code false} if, on
     * restart, the I/O exit expects the source to be opened at the
     * offset that the source agent intends to start reading from (the
     * {@link IOExitPath#openForRead(long)} method can be invoked with a
     * non-zero value as its argument).
     */
    public boolean getRereadSourceOnRestart() {
        return rereadSourceOnRestart;
    }

    /**
     * Sets the value to determine whether the I/O exit implementation expects
     * the resource to be re-read from the beginning if a transfer is restarted.
     * <p>
     * The default is {@code true}. The I/O exit should call this method when
     * required to change this value.
     *
     * @param rereadSourceOnRestart
     *        {@code true} if, on restart, the I/O exit expects the source
     * resource to be opened at the beginning and re-read from the
     * beginning (the {@link IOExitPath#openForRead(long)} method
     * is always invoked with 0L as an argument). {@code false}
     * if, on restart, the I/O exit expects the source to be opened
     * at the offset that the source agent intends to start reading
     * from (the {@link IOExitPath#openForRead(long)} method can be
     * invoked with a non-zero value as its argument).
     */
    public void setRereadSourceOnRestart(boolean rereadSourceOnRestart) {
        this.rereadSourceOnRestart = rereadSourceOnRestart;
    }

    /**
     * Determines whether the I/O exit implementation requires the source
     * resource to be re-checksummed if the transfer is restarted.
     * Re-checksumming takes place only if the
     * {@link #getRereadSourceOnRestart()} method returns {@code true}.
     *
     * @return {@code true} if, on restart, the I/O exit expects the already-
     * transferred portion of the source to be re-checksummed for
     * inconsistencies. Use this option in environments
     * where the source could be changed during a restart. {@code
     * false} if, on restart, the I/O exit does not require the
     * already-transferred portion of the source to be re-checksummed.
     */
}
```

```

public boolean getRechecksumSourceOnRestart() {
    return rechecksumSourceOnRestart;
}

/**
 * Sets the value to determine whether the I/O exit implementation requires
 * the source resource to be re-checksummed if the transfer is restarted.
 * Re-checksumming takes place only if the
 * {@link #getRereadSourceOnRestart()} method returns {@code true}.
 * <p>
 * The default is {@code true}. The I/O exit should call this method when
 * required to change this value.
 *
 * @param rechecksumSourceOnRestart
 *     {@code true} if, on restart, the I/O exit expects the already
 *     transferred portion of the source to be re-checksummed
 *     for inconsistencies. Use this option in environments
 *     where the source could be changed during a restart.
 *     {@code false} if, on restart, the I/O exit does not
 *     require the already-transferred portion of the source to be
 *     re-checksummed.
 */
public void setRechecksumSourceOnRestart(boolean rechecksumSourceOnRestart) {
    this.rechecksumSourceOnRestart = rechecksumSourceOnRestart;
}

/**
 * Determines whether the I/O exit implementation requires the destination
 * resource to be re-checksummed if the transfer is restarted.
 *
 * @return {@code true} if, on restart, the I/O exit expects the already
 *     transferred portion of the destination to be re-checksummed to
 *     check for inconsistencies. This option should be used in
 *     environments where the destination could have been changed while
 *     a restart is occurring. {@code false} if, on restart, the I/O exit
 *     does not require the already transferred portion of the
 *     destination to be re-checksummed.
 */
public boolean getRechecksumDestinationOnRestart() {
    return rechecksumDestinationOnRestart;
}

/**
 * Sets the value to determine whether the I/O exit implementation requires
 * the destination resource to be re-checksummed if the transfer is
 * restarted.
 * <p>
 * The default is {@code true}. The I/O exit should call this method when
 * required to change this value.
 *
 * @param rechecksumDestinationOnRestart
 *     {@code true} if, on restart, the I/O exit expects the already-
 *     transferred portion of the destination to be re-checksummed
 *     for inconsistencies. Use this option in environments
 *     where the destination could have been changed during a
 *     restart. {@code false} if, on restart, the I/O exit does not
 *     require the already-transferred portion of the destination
 *     to be re-checksummed.
 */
public void setRechecksumDestinationOnRestart(
    boolean rechecksumDestinationOnRestart) {
    this.rechecksumDestinationOnRestart = rechecksumDestinationOnRestart;
}

/**
 * Determines whether the I/O exit implementation requires the use of an
 * intermediate file when writing the data at the destination. The
 * intermediate file mechanism is typically used to prevent an incomplete
 * destination resource from being processed.
 *
 * @return {@code true} if data should be written to an intermediate file at
 *     the destination and then renamed (to the requested destination
 *     path name as specified in the transfer request) after the transfer is
 *     complete. {@code false} if data should be written directly to the
 *     requested destination path name without the use of an
 *     intermediate file.
 */
public boolean getUseIntermediateFileAtDestination() {
    return useIntermediateFileAtDestination;
}

```



```

* Sets the value to determine whether the I/O exit implementation requires
* the use of an intermediate file when writing the data at the destination.
* The intermediate file mechanism is typically used to prevent an
* incomplete destination resource from being processed.
*
* <p>
* The default is {@code true}. The I/O exit should call this method when
* required to change this value.
*
* @param useIntermediateFileAtDestination
*     {@code true} if data should be written to an intermediate file
*     at the destination and then renamed (to the requested
*     destination path name as specified in the transfer request) after
*     the transfer is complete. {@code false} if data should be written
*     directly to the requested destination path name without the
*     use of an intermediate file
*/
public void setUseIntermediateFileAtDestination(
    boolean useIntermediateFileAtDestination) {
    this.useIntermediateFileAtDestination = useIntermediateFileAtDestination;
}

/**
* Determines whether the I/O exit implementation requires
* {@link IOExitChannel} instances to be accessed by a single thread only.
*
* @return {@code true} if {@link IOExitChannel} instances are to be
*     accessed by a single thread only.
*/
public boolean requiresSingleThreadedChannelIO() {
    return requiresSingleThreadedChannelIO;
}

/**
* Sets the value to determine whether the I/O exit implementation requires
* channel operations for a particular instance to be accessed by a
* single thread only.
* <p>
* For certain I/O implementations it is necessary that resource path
* operations such as open, read, write, and close are invoked only from a
* single execution {@link Thread}. When set {@code true}, WMQFTE ensures
* that the following are invoked on a single thread:
* <ul>
* <li>{@link IOExitResourcePath#openForRead(long)} method and all methods of
* the returned {@link IOExitChannel} instance.</li>
* <li>{@link IOExitResourcePath#openForWrite(boolean)} method and all
* methods of the returned {@link IOExitChannel} instance.</li>
* </ul>
* <p>
* This has a slight performance impact, hence enable single-threaded channel
* I/O only when absolutely necessary.
* <p>
* The default is {@code false}. The I/O exit should call this method when
* required to change this value.
*
* @param requiresSingleThreadedChannelIO
*     {@code true} if {@link IOExitChannel} instances are to be
*     accessed by a single thread only.
*/
public void setRequiresSingleThreadedChannelIO(boolean requiresSingleThreadedChannelIO) {
    this.requiresSingleThreadedChannelIO = requiresSingleThreadedChannelIO;
}
}
}

```

## Tarefas relacionadas

[Usando as saídas de usuário de E/S de transferência do MFT](#)

[Customizando o MFT com saídas de usuário](#)

## Interface *IOExitRecordChannel.java*

### IOExitRecordChannel.java

```

/*
* Licensed Materials - Property of IBM
*
* "Restricted Materials of IBM"

```

```

*
* 5724-H72
*
*   Copyright IBM Corp. 2011, 2024. All Rights Reserved.
*
* US Government Users Restricted Rights - Use, duplication or
* disclosure restricted by GSA ADP Schedule Contract with
* IBM Corp.
*/
package com.ibm.wmqfte.exitroutine.api;

import java.io.IOException;
import java.nio.ByteBuffer;

/**
 * Represents a channel that enables records of data to be read from or written
 * to an {@link IOExitRecordResourcePath} resource.
 * <p>
 * This is an extension of the {@link IOExitChannel} interface such that the
 * {@link #read(java.nio.ByteBuffer)} and {@link #write(java.nio.ByteBuffer)}
 * methods are expected to deal in whole records of data only. That is, the
 * {@link java.nio.ByteBuffer} returned from the read method and passed to the
 * write method is assumed to contain one or more complete records.
 */
public interface IOExitRecordChannel extends IOExitChannel {

    /**
     * Reads records from this channel into the given buffer, starting at this
     * channel's current position, and updates the current position by the
     * amount of data read.
     * <p>
     * Record data is copied into the buffer starting at its current position
     * and up to its limit. On return, the buffer's position is updated to
     * reflect the number of bytes read.
     * <p>
     * Only whole records are copied into the buffer.
     * <p>
     * For a fixed-record-format resource, this might be multiple records. The
     * amount of data in the return buffer does not necessarily need to be a
     * multiple of the record length, but the last record is still to be treated
     * as a complete record and padded as required by the caller.
     * <p>
     * For a variable-format resource, this is a single whole record of a size
     * corresponding to the amount of return data or multiple whole records with
     * all except the last being treated as records of maximum size.
     *
     * @param buffer
     *     The buffer that the record data is to be copied into.
     * @return
     *     The number of bytes read, which might be zero, or -1 if the end of
     *     data has been reached.
     * @throws RecoverableIOException
     *     If a recoverable problem occurs while reading the data. For a
     *     WMQFTE transfer this means that it will attempt to recover.
     * @throws IOException
     *     If some other I/O problem occurs, for example, if the passed
     *     buffer is insufficient to contain at least one complete
     *     record). For a WMQFTE transfer this means that it will be
     *     failed.
     */
    int read(ByteBuffer buffer) throws RecoverableIOException, IOException;

    /**
     * Writes records to this channel from the given buffer, starting at this
     * channel's current position, and updates the current position by the
     * amount of data written. The channel's resource is grown to accommodate
     * the data, if necessary.
     * <p>
     * Record data is copied from the buffer starting at its current position
     * and up to its limit. On return, the buffer's position is updated to
     * reflect the number of bytes written.
     * <p>
     * The buffer is expected to contain only whole records.
     * <p>
     * For a fixed-record-format resource, this might be multiple records and if
     * there is insufficient data in the buffer for a complete record, the
     * record is to be padded as required to complete the record.
     * <p>
     * For a variable-record format resource the buffer is normally expected to
     * contain a single record of length corresponding to the amount of data
     * within the buffer. However, if the amount of data within the buffer
     * exceeds the maximum record length, the implementation can either:
     * <ol>

```

```

* <li>throw an {@link IOException} indicating that it cannot handle the
* situation.</li>
* <li>Consume a record's worth of data from the buffer, leaving the remaining
* data within the buffer.</li>
* <li>Consume all the buffer data and just write what it can to the current
* record. This effectively truncates the data.</li>
* <li>Consume all the buffer data and write to multiple records.</li>
* </ol>
*
* @param buffer
*         The buffer containing the data to be written.
* @return The number of bytes written, which might be zero.
* @throws RecoverableIOException
*         If a recoverable problem occurs while writing the data. For a
*         WMQFTE transfer this means that it will attempt to recover.
* @throws IOException
*         If some other I/O problem occurs. For a WMQFTE transfer this
*         means that it will be failed.
*/
int write(ByteBuffer buffer) throws RecoverableIOException, IOException;
}

```

### Tarefas relacionadas

[Usando as saídas de usuário de E/S de transferência do MFT](#)

[Customizando o MFT com saídas de usuário](#)

## **Interface IOExitRecordResourcePath.java**

### IOExitRecordResourcePath.java

```

/*
 * Licensed Materials - Property of IBM
 *
 * "Restricted Materials of IBM"
 *
 * 5724-H72
 *
 * © Copyright IBM Corp. 2011, 2024. All Rights Reserved.
 *
 * US Government Users Restricted Rights - Use, duplication or
 * disclosure restricted by GSA ADP Schedule Contract with
 * IBM Corp.
 */
package com.ibm.wmqfte.exitroutine.api;

import java.io.IOException;

/**
 * Represents a path that denotes a record-oriented data resource (for example,
 * a z/OS data set). It allows the data to be located, the record format to be
 * understood, and {@link IOExitRecordChannel} instances to be created for read
 * or write operations.
 */
public interface IOExitRecordResourcePath extends IOExitResourcePath {

    /**
     * Record formats for record-oriented resources.
     */
    public enum RecordFormat {
        FIXED, VARIABLE
    }

    /**
     * Obtains the record length for records that are maintained by the resource
     * denoted by this abstract path.
     * <p>
     * For a resource with fixed-length records, the data for each record read
     * and written is assumed to be this length.
     * <p>
     * For a resource with variable-length records, this is the maximum length
     * for a record's data.
     * <p>
     * This method should return a value greater than zero, otherwise it can
     * result in the failure of a WMQFTE transfer that involves this abstract
     * path.
     */
}

```

```

*
* @return The record length, in bytes, for records maintained by the
*         resource.
*/
int getRecordLength();

/**
 * Obtains record format, as a {@link RecordFormat} instance, for records
 * that are maintained by the resource denoted by this abstract path.
 *
 * @return A {@link RecordFormat} instance for the record format for records
 *         that are maintained by the resource denoted by this abstract
 *         path.
 */
RecordFormat getRecordFormat();

/**
 * Opens a {@link IOExitRecordChannel} instance for reading data from the
 * resource denoted by this abstract path. The current data byte position
 * for the resource is expected to be the passed position value, such that
 * when {@link IOExitRecordChannel#read(java.nio.ByteBuffer)} is called,
 * data starting from that position is read.
 * <p>
 * Note that the data byte read position will be on a record boundary.
 *
 * @param position
 *         The required data byte read position.
 * @return A new {@link IOExitRecordChannel} instance allowing data to be
 *         read from the resource denoted by this abstract path.
 * @throws RecoverableIOException
 *         If a recoverable problem occurs while attempting to open the
 *         resource for reading. This means that WMQFTE can attempt to
 *         recover the transfer.
 * @throws IOException
 *         If some other I/O problem occurs.
 */
IOExitRecordChannel openForRead(long position)
    throws RecoverableIOException, IOException;

/**
 * Opens a {@link IOExitRecordChannel} instance for writing data to the
 * resource denoted by this abstract path. Writing of data, using the
 * {@link IOExitRecordChannel#write(java.nio.ByteBuffer)} method, starts at
 * either the beginning of the resource or end of the current data for the
 * resource, depending on the specified append parameter.
 *
 * @param append
 *         When {@code true} indicates that data written to the resource
 *         should be appended to the end of the current data. When
 *         {@code false} indicates that writing of data is to start at
 *         the beginning of the resource; any existing data is lost.
 * @return A new {@link IOExitRecordChannel} instance allowing data to be
 *         written to the resource denoted by this abstract path.
 * @throws RecoverableIOException
 *         If a recoverable problem occurs while attempting to open the
 *         resource for writing. This means that WMQFTE can attempt to
 *         recover the transfer.
 * @throws IOException
 *         If some other I/O problem occurs.
 */
IOExitRecordChannel openForWrite(boolean append)
    throws RecoverableIOException, IOException;
}

```

## Tarefas relacionadas

[Usando as saídas de usuário de E/S de transferência do MFT](#)

[Customizando o MFT com saídas de usuário](#)

## Interface *IOExitResourcePath.java*

### IOExitResourcePath.java

```

/*
 * Licensed Materials - Property of IBM
 *
 * "Restricted Materials of IBM"

```

```

*
* 5724-H72
*
*   Copyright IBM Corp. 2011, 2024. All Rights Reserved.
*
* US Government Users Restricted Rights - Use, duplication or
* disclosure restricted by GSA ADP Schedule Contract with
* IBM Corp.
*/
package com.ibm.wmqfte.exitroutine.api;

import java.io.IOException;

/**
 * Represents a path that denotes a data resource (for example, a file,
 * directory, or group of database records). It allows the data to be located
 * and {@link IOExitChannel} instances to be created for read or write
 * operations.
 * <p>
 * There are two types of data resources as follows:
 * <ul>
 * <li>Directory - a container for other data resources. The
 * {@link #isDirectory()} method returns {@code true} for these.</li>
 * <li>File - a data container. This allows data to be read from or written to
 * it. The {@link #isFile()} method returns {@code true} for these.</li>
 * </ul>
 */
public interface IOExitResourcePath extends IOExitPath {

    /**
     * Creates a new {@link IOExitResourcePath} instance for a child path of the
     * resource denoted by this abstract path.
     * <p>
     * For example, with a UNIX-style path, {@code
     * IOExitResourcePath("/home/fteuser/test").newPath("subtest")} could be
     * equivalent to: {@code IOExitResourcePath("/home/fteuser/test/subtest")}
     *
     * @param child
     *         The child path name.
     * @return A new {@link IOExitResourcePath} instance that represents a child
     *         of this path.
     */
    IOExitResourcePath newPath(final String child);

    /**
     * Creates the directory path for the resource denoted by this abstract
     * path, including any necessary but nonexistent parent directories. If the
     * directory path already exists, this method has no effect.
     * <p>
     * If this operation fails, it might have succeeded in creating some of the
     * necessary parent directories.
     *
     * @throws IOException
     *         If the directory path cannot be fully created, when it does
     *         not already exist.
     */
    void makePath() throws IOException;

    /**
     * Obtains the canonical path of the abstract path as a {@link String}.
     * <p>
     * A canonical path is defined as being absolute and unique. For example,
     * the path can be represented as UNIX-style relative path: {@code
     * test/file.txt} but the absolute and unique canonical path representation
     * is: {@code /home/fteuser/test/file.txt}
     *
     * @return The canonical path as a {@link String}.
     * @throws IOException
     *         If the canonical path cannot be determined for any reason.
     */
    String getCanonicalPath() throws IOException;

    /**
     * Tests if this abstract path is an absolute path.
     * <p>
     * For example, a UNIX-style path, {@code /home/fteuser/test} is an absolute
     * path, whereas {@code fteuser/test} is not.
     *
     * @return {@code true} if this abstract path is an absolute path, {@code
     *         false} otherwise.
     */
    boolean isAbsolute();

```

```

/**
 * Tests if the resource denoted by this abstract path exists.
 *
 * @return {@code true} if the resource denoted by this abstract path
 *         exists, {@code false} otherwise.
 * @throws IOException
 *         If the existence of the resource cannot be determined for any
 *         reason.
 */
boolean exists() throws IOException;

/**
 * Tests whether the calling application can read the resource denoted by
 * this abstract path.
 *
 * @return {@code true} if the resource for this path exists and can be
 *         read, {@code false} otherwise.
 * @throws IOException
 *         If a problem occurs while attempting to determine if the
 *         resource can be read.
 */
boolean canRead() throws IOException;

/**
 * Tests whether the calling application can modify the resource denoted by
 * this abstract path.
 *
 * @return {@code true} if the resource for this path exists and can be
 *         modified, {@code false} otherwise.
 * @throws IOException
 *         If a problem occurs while attempting to determine if the
 *         resource can be modified.
 */
boolean canWrite() throws IOException;

/**
 * Tests whether the specified user is permitted to read the resource
 * denoted by this abstract path.
 * <p>
 * When WMQFTE invokes this method, the user identifier is the MQMD user
 * identifier for the requesting transfer.
 *
 * @param userId
 *         User identifier to test for access.
 * @return {@code true} if the resource for this abstract path exists and is
 *         permitted to be read by the specified user, {@code false}
 *         otherwise.
 * @throws IOException
 *         If a problem occurs while attempting to determine if the user
 *         is permitted to read the resource.
 */
boolean readPermitted(String userId) throws IOException;

/**
 * Tests whether the specified user is permitted to modify the resource
 * denoted by this abstract path.
 * <p>
 * When WMQFTE invokes this method, the user identifier is the MQMD user
 * identifier for the requesting transfer.
 *
 * @param userId
 *         User identifier to test for access.
 * @return {@code true} if the resource for this abstract path exists and is
 *         permitted to be modified by the specified user, {@code false}
 *         otherwise.
 * @throws IOException
 *         If a problem occurs while attempting to determine if the user
 *         is permitted to modify the resource.
 */
boolean writePermitted(String userId) throws IOException;

/**
 * Tests if the resource denoted by this abstract path is a directory-type
 * resource.
 *
 * @return {@code true} if the resource denoted by this abstract path is a
 *         directory type resource, {@code false} otherwise.
 */
boolean isDirectory();
/**

```

```

* Creates the resource denoted by this abstract path, if it does not
* already exist.
*
* @return {@code true} if the resource does not exist and was successfully
*         created, {@code false} if the resource already existed.
* @throws RecoverableIOException
*         If a recoverable problem occurs while attempting to create
*         the resource. This means that WMQFTE can attempt to recover
*         the transfer.
* @throws IOException
*         If some other I/O problem occurs.
*/
boolean createNewPath() throws RecoverableIOException, IOException;

/**
* Tests if the resource denoted by this abstract path is a file-type
* resource.
*
* @return {@code true} if the resource denoted by this abstract path is a
*         file type resource, {@code false} otherwise.
*/
boolean isFile();

/**
* Obtains the last modified time for the resource denoted by this abstract
* path.
* <p>
* This time is measured in milliseconds since the epoch (00:00:00 GMT,
* January 1, 1970).
*
* @return The last modified time for the resource denoted by this abstract
*         path, or a value of 0L if the resource does not exist or a
*         problem occurs.
*/
long lastModified();

/**
* Deletes the resource denoted by this abstract path.
* <p>
* If the resource is a directory, it must be empty for the delete to work.
*
* @throws IOException
*         If the delete of the resource fails for any reason.
*/
void delete() throws IOException;

/**
* Renames the resource denoted by this abstract path to the specified
* destination abstract path.
* <p>
* The rename should still be successful if the resource for the specified
* destination abstract path already exists and it is possible to replace
* it.
*
* @param destination
*         The new abstract path for the resource denoted by this
*         abstract path.
* @throws IOException
*         If the rename of the resource fails for any reason.
*/
void renameTo(IOExceptionResourcePath destination) throws IOException;

/**
* Creates a new path to use for writing to a temporary resource that did
* not previously exist.
* <p>
* The implementation can choose the abstract path name for the temporary
* resource. However, for clarity and problem diagnosis, the abstract path
* name for the temporary resource should be based on this abstract path
* name with the specified suffix appended and additional characters to make
* the path unique (for example, sequence numbers), as required.
* <p>
* When WMQFTE transfers data to a destination it normally attempts to first
* write to a temporary resource then on transfer completion renames the
* temporary resource to the required destination. This method is called by
* WMQFTE to create a new temporary resource path. The returned path should
* be new and the resource should not previously exist.
*
* @param suffix
*         Recommended suffix to use for the generated temporary path.
*
* @return A new {@link IOExceptionResourcePath} instance for the temporary

```

```

*         resource path, that did not previously exist.
* @throws RecoverableIOException
*         If a recoverable problem occurs whilst attempting to create
*         the temporary resource. This means that WMQFTE can attempt to
*         recover the transfer.
* @throws IOException
*         If some other I/O problem occurs.
*/
IOExitResourcePath createTempPath(String suffix)
    throws RecoverableIOException, IOException;

/**
 * Opens a {@link IOExitChannel} instance for reading data from the resource
 * denoted by this abstract path. The current data byte position for the
 * resource is expected to be the passed position value, such that when
 * {@link IOExitChannel#read(java.nio.ByteBuffer)} is called, data starting
 * from that position is read.
 *
 * @param position
 *         The required data byte read position.
 * @return A new {@link IOExitChannel} instance allowing data to be read
 *         from the resource denoted by this abstract path.
 * @throws RecoverableIOException
 *         If a recoverable problem occurs while attempting to open the
 *         resource for reading. This means that WMQFTE can attempt to
 *         recover the transfer.
 * @throws IOException
 *         If some other I/O problem occurs.
 */
IOExitChannel openForRead(long position) throws RecoverableIOException,
    IOException;

/**
 * Opens a {@link IOExitChannel} instance for writing data to the resource
 * denoted by this abstract path. Writing of data, using the
 * {@link IOExitChannel#write(java.nio.ByteBuffer)} method, starts at either
 * the beginning of the resource or end of the current data for the
 * resource, depending on the specified append parameter.
 *
 * @param append
 *         When {@code true} indicates that data written to the resource
 *         should be appended to the end of the current data. When
 *         {@code false} indicates that writing of data is to start at
 *         the beginning of the resource; any existing data is lost.
 * @return A new {@link IOExitChannel} instance allowing data to be written
 *         to the resource denoted by this abstract path.
 * @throws RecoverableIOException
 *         If a recoverable problem occurs whilst attempting to open the
 *         resource for writing. This means that WMQFTE can attempt to
 *         recover the transfer.
 * @throws IOException
 *         If some other I/O problem occurs.
 */
IOExitChannel openForWrite(boolean append) throws RecoverableIOException,
    IOException;

/**
 * Tests if the resource denoted by this abstract path is in use by another
 * application. Typically, this is because another application has a lock on
 * the resource either for shared or exclusive access.
 *
 * @return {@code true} if resource denoted by this abstract path is in use
 *         by another application, {@code false} otherwise.
 */
boolean inUse();

/**
 * Obtains a {@link IOExitProperties} instance for properties associated
 * with the resource denoted by this abstract path.
 *
 * <p>
 * WMQFTE will read these properties to govern how a transfer behaves when
 * interacting with the resource.
 *
 * @return A {@link IOExitProperties} instance for properties associated
 *         with the resource denoted by this abstract path.
 */
IOExitProperties getProperties();
}

```



## Tarefas relacionadas

[Usando as saídas de usuário de E/S de transferência do MFT](#)

[Customizando o MFT com saídas de usuário](#)

## Interface *IOExitWildcardPath.java*

### IOExitWildcardPath.java

```
/*
 * Licensed Materials - Property of IBM
 *
 * "Restricted Materials of IBM"
 *
 * 5724-H72
 *
 * Copyright IBM Corp. 2011, 2024. All Rights Reserved.
 *
 * US Government Users Restricted Rights - Use, duplication or
 * disclosure restricted by GSA ADP Schedule Contract with
 * IBM Corp.
 */
package com.ibm.wmqfte.exitroutine.api;

/**
 * Represents a path that denotes a wildcard. This can be used to match multiple
 * resource paths.
 */
public interface IOExitWildcardPath extends IOExitPath {
```

## Tarefas relacionadas

[Usando as saídas de usuário de E/S de transferência do MFT](#)

[Customizando o MFT com saídas de usuário](#)

## Interface *MonitorExit.java*

### MonitorExit.java

```
/*
 * Licensed Materials - Property of IBM
 *
 * "Restricted Materials of IBM"
 *
 * 5724-H72
 *
 * Copyright IBM Corp. 2009, 2024. All Rights Reserved.
 *
 * US Government Users Restricted Rights - Use, duplication or
 * disclosure restricted by GSA ADP Schedule Contract with
 * IBM Corp.
 */
package com.ibm.wmqfte.exitroutine.api;

import java.util.Map;

/**
 * An interface that is implemented by classes that want to be invoked as part of
 * user exit routine processing. This interface defines a method that will be
 * invoked immediately prior to starting a task as the result of a monitor trigger
 */
public interface MonitorExit {

    /**
     * Invoked immediately prior to starting a task as the result of a monitor
     * trigger.
     *
     * @param environmentMetaData
     * meta data about the environment in which the implementation
     * of this method is running. This information can only be read,
     * it cannot be updated by the implementation. The constant
     * defined in EnvironmentMetaDataConstants class can
     * be used to access the data held by this map.
     */
}
```

```

*
* @param monitorMetaData
*         meta data to associate with the monitor. The meta data passed
*         to this method can be altered, and the changes will be
*         reflected in subsequent exit routine invocations. This map
*         also contains keys with IBM reserved names. These entries are
*         defined in the <code>MonitorMetaDataConstants</code> class and
*         have special semantics. The the values of the IBM reserved names
*         cannot be modified by the exit
*
* @param taskDetails
*         An XML String representing the task to be executed as a result of
*         the monitor triggering. This XML string may be modified by the
*         exit
*
* @return  a monitor exit result object which is used to determine if the
*         task should proceed, or be cancelled.
*/
MonitorExitResult onMonitor(Map<String, String> environmentMetaData,
                             Map<String, String> monitorMetaData,
                             Reference<String> taskDetails);
}

```

## Tarefas relacionadas

[Monitorando recursos do MFT](#)

[Customizando o MFT com saídas de usuário](#)

## Referências relacionadas

[“Interface SourceTransferStartExit.java” na página 2235](#)

[“Interface SourceTransferEndExit.java” na página 2234](#)

[“Interface DestinationTransferStartExit.java” na página 2211](#)

[“Interface DestinationTransferEndExit.java” na página 2209](#)

[“Interface ProtocolBridgeCredentialExit.java” na página 2230](#)

## ***Interface ProtocolBridgeCredentialExit.java***

### **ProtocolBridgeCredentialExit.java**

```

/*
 * Licensed Materials - Property of IBM
 *
 * "Restricted Materials of IBM"
 *
 * 5724-H72
 *
 * © Copyright IBM Corp. 2008, 2024. All Rights Reserved.
 *
 * US Government Users Restricted Rights - Use, duplication or
 * disclosure restricted by GSA ADP Schedule Contract with
 * IBM Corp.
 */
package com.ibm.wmqfte.exitroutine.api;

import java.util.Map;

/**
 * An interface that is implemented by classes that are to be invoked as part of
 * user exit routine processing. This interface defines methods that will
 * be invoked by a protocol bridge agent to map the MQ user ID of the transfer to credentials
 * that are to be used to access the protocol server.
 * There will be one instance of each implementation class per protocol bridge agent. The methods
 * can be called from different threads so the methods must be synchronized.
 */
public interface ProtocolBridgeCredentialExit {

    /**
     * Invoked once when a protocol bridge agent is started. It is intended to initialize
     * any resources that are required by the exit
     */
    * @param bridgeProperties
    *         The values of properties defined for the protocol bridge.
    *         These values can only be read, they cannot be updated by

```

```

*         the implementation.
*
*
* @return true if the initialization is successful and false if unsuccessful
*         If false is returned from an exit the protocol bridge agent will not
*         start
*/

public boolean initialize(final Map<String> bridgeProperties);

/**
 * Invoked once for each transfer to map the MQ user ID in the transfer message to the
 * credentials to be used to access the protocol server
 *
 * @param mqUserId The MQ user ID from which to map to the credentials to be used
 *                access the protocol server
 * @return A credential exit result object that contains the result of the map and
 *         the credentials to use to access the protocol server
 */

public CredentialExitResult mapMQUserId(final String mqUserId);

/**
 * Invoked once when a protocol bridge agent is shutdown. It is intended to release
 * any resources that were allocated by the exit
 *
 * @param bridgeProperties
 *       The values of properties defined for the protocol bridge.
 *       These values can only be read, they cannot be updated by
 *       the implementation.
 *
 * @return
 */
*/

public void shutdown(final Map<String> bridgeProperties);
}

```

## Tarefas relacionadas

[Customizando o MFT com saídas de usuário](#)

[Mapeando Credenciais para um Servidor de Arquivos Usando Classes de Saída](#)

## ***Interface ProtocolBridgeCredentialExit2.java***

### **ProtocolBridgeCredentialExit2.java**

```

/*
 * Licensed Materials - Property of IBM
 *
 * "Restricted Materials of IBM"
 *
 * 5724-H72
 *
 * © Copyright IBM Corp. 2011, 2024. All Rights Reserved.
 *
 * US Government Users Restricted Rights - Use, duplication or
 * disclosure restricted by GSA ADP Schedule Contract with
 * IBM Corp.
 */
package com.ibm.wmqfte.exitroutine.api;

/**
 * An interface that is implemented by classes that are invoked as part of user
 * exit routine processing. This interface defines methods that are invoked by a
 * protocol bridge agent to map the MQ user ID of the transfer to credentials
 * used to access a specified protocol bridge server. There will be one instance
 * of each implementation class for each protocol bridge agent. The methods can
 * be called from different threads so the methods must be synchronized.
 */
public interface ProtocolBridgeCredentialExit2 extends
    ProtocolBridgeCredentialExit {

    /**
     * Invoked once for each transfer to map the MQ user ID in the transfer
     * message to the credentials used to access a specified protocol server.
     *
     */

```

```

    * @param endPoint
    *         Information that describes the protocol server to be accessed.
    * @param mqUserId
    *         The MQ user ID from which to map the credentials used to
    *         access the protocol server.
    * @return A {@link CredentialExitResult} instance that contains the result
    *         of the map and the credentials to use to access the protocol
    *         server.
    */
    public CredentialExitResult mapMQUserId(
        final ProtocolServerEndPoint endPoint, final String mqUserId);
}

```

## Tarefas relacionadas

[Customizando o MFT com saídas de usuário](#)

[Mapeando Credenciais para um Servidor de Arquivos Usando Classes de Saída](#)

## ***Interface ProtocolBridgePropertiesExit2.java***

### **ProtocolBridgePropertiesExit2.java**

```

/**
 * Licensed Materials - Property of IBM
 *
 * "Restricted Materials of IBM"
 *
 * 5724-H72
 *
 * © Copyright IBM Corp. 2011, 2024. All Rights Reserved.
 *
 * US Government Users Restricted Rights - Use, duplication or
 * disclosure restricted by GSA ADP Schedule Contract with
 * IBM Corp.
 */
package com.ibm.wmqfte.exitroutine.api;

import java.util.Map;
import java.util.Properties;

/**
 * An interface that is implemented by classes that are to be invoked as part of
 * user exit routine processing. This interface defines methods that will be
 * invoked by a protocol bridge agent to look up properties for protocol servers
 * that are referenced in transfers.
 * <p>
 * There will be one instance of each implementation class for each protocol
 * bridge agent. The methods can be called from different threads so the methods
 * must be synchronised.
 */
public interface ProtocolBridgePropertiesExit2 {

    /**
     * Invoked once when a protocol bridge agent is started. It is intended to
     * initialize any resources that are required by the exit.
     *
     * @param bridgeProperties
     *         The values of properties defined for the protocol bridge.
     *         These values can only be read, they cannot be updated by the
     *         implementation.
     * @return {@code true} if the initialization is successful and {@code
     *         false} if unsuccessful. If {@code false} is returned from an exit
     *         the protocol bridge agent will not start.
     */
    public boolean initialize(final Map<String, String> bridgeProperties);

    /**
     * Invoked when the Protocol Bridge needs to access the protocol bridge credentials XML file.
     *
     * @return a {@link String} object giving the location of the ProtocolBridgeCredentials.xml
     */
    public String getCredentialLocation ();

    /**
     * Obtains a set of properties for the specified protocol server name.
     * <p>
     * The returned {@link Properties} must contain entries with key names

```

```

* corresponding to the constants defined in
* {@link ProtocolServerPropertyConstants} and in particular must include an
* entry for all appropriate constants described as required.
*
* @param protocolServerName
*     The name of the protocol server whose properties are to be
*     returned. If a null or a blank value is specified, properties
*     for the default protocol server are to be returned.
* @return The {@link Properties} for the specified protocol server, or null
*     if the server cannot be found.
*/
public Properties getProtocolServerProperties(
    final String protocolServerName);

/**
* Invoked once when a protocol bridge agent is shut down. It is intended to
* release any resources that were allocated by the exit.
*
* @param bridgeProperties
*     The values of properties defined for the protocol bridge.
*     These values can only be read, they cannot be updated by the
*     implementation.
*/
public void shutdown(final Map<String, String> bridgeProperties);
}

```

### Tarefas relacionadas

[ProtocolBridgePropertiesExit: consultando as propriedades do servidor de arquivos de protocolo](#)

[Customizando o MFT com saídas de usuário](#)

[Mapeando Credenciais para um Servidor de Arquivos Usando Classes de Saída](#)

### Classe *SourceFileExitFileSpecification.java*

#### SourceFileExitFileSpecification.java

```

/*
* Licensed Materials - Property of IBM
*
* "Restricted Materials of IBM"
*
* 5724-H72
*
* © Copyright IBM Corp. 2012, 2024. All Rights Reserved.
*
* US Government Users Restricted Rights - Use, duplication or
* disclosure restricted by GSA ADP Schedule Contract with
* IBM Corp.
*/
package com.ibm.wmqfte.exitroutine.api;

import java.util.Map;

/**
* A specification of the file names to use for a file transfer, as evaluated by the
* agent acting as the source of the transfer.
*/
public final class SourceFileExitFileSpecification {

    private final String sourceFileSpecification;
    private final String destinationFileSpecification;
    private final Map<String, String> sourceFileMetaData;
    private final Map<String, String> destinationFileMetaData;

    /**
    * Constructor. Creates a source file exit file specification.
    *
    * @param sourceFileSpecification
    *     the source file specification to associate with the source file
    *     exit file specification.
    *
    * @param destinationFileSpecification
    *     the destination file specification to associate with the
    *     source file exit file specification.
    *
    * @param sourceFileMetaData
    */
}

```

```

*         the source file meta data.
*
* @param destinationFileMetaData
*         the destination file meta data
*/
public SourceFileExitFileSpecification(final String sourceFileSpecification,
                                       final String destinationFileSpecification,
                                       final Map<String, String> sourceFileMetaData,
                                       final Map<String, String> destinationFileMetaData) {
    this.sourceFileSpecification = sourceFileSpecification;
    this.destinationFileSpecification = destinationFileSpecification;
    this.sourceFileMetaData = sourceFileMetaData;
    this.destinationFileMetaData = destinationFileMetaData;
}

/**
 * Returns the destination file specification.
 *
 * @return the destination file specification. This represents the location,
 *         on the agent acting as the destination for the transfer, where the
 *         file should be written. Exit routines installed into the agent
 *         acting as the destination for the transfer may override this value.
 */
public String getDestination() {
    return destinationFileSpecification;
}

/**
 * Returns the source file specification.
 *
 * @return the source file specification. This represents the location where
 *         the file data will be read from.
 */
public String getSource() {
    return sourceFileSpecification;
}

/**
 * Returns the file meta data that relates to the source file specification.
 *
 * @return the file meta data that relates to the source file specification.
 */
public Map<String, String> getSourceFileMetaData() {
    return sourceFileMetaData;
}

/**
 * Returns the file meta data that relates to the destination file specification.
 *
 * @return the file meta data that relates to the destination file specification.
 */
public Map<String, String> getDestinationFileMetaData() {
    return destinationFileMetaData;
}
}

```

## Conceitos relacionados

[“Metadados para saídas de usuário do MFT” na página 2196](#)

Há três tipos diferentes de metadados que podem ser fornecidos para rotinas de saída do usuário para o Managed File Transfer: metadados de ambiente, de transferência e de arquivo. Esses metadados são apresentados como mapas de pares de valores e chave Java.

## Interface *SourceTransferEndExit.java*

### SourceTransferEndExit.java

```

/*
 * Licensed Materials - Property of IBM
 *
 * "Restricted Materials of IBM"
 *
 * 5724-H72
 *
 * © Copyright IBM Corp. 2008, 2024. All Rights Reserved.
 *
 * US Government Users Restricted Rights - Use, duplication or

```

```

* disclosure restricted by GSA ADP Schedule Contract with
* IBM Corp.
*/
package com.ibm.wmqfte.exitpoint.api;

/**
 * An interface that is implemented by classes that want to be invoked as part of
 * user exit routine processing. This interface defines a method that will be
 * invoked immediately after completing a transfer on the agent acting as the
 * source of the transfer.
 */
public interface SourceTransferEndExit {

    /**
     * Invoked immediately after the completion of a transfer on the agent acting as
     * the source of the transfer.
     *
     * @param transferExitResult
     *        a result object reflecting whether or not the transfer completed
     *        successfully.
     *
     * @param sourceAgentName
     *        the name of the agent acting as the source of the transfer.
     *        This is the name of the agent that the implementation of this
     *        method will be invoked from.
     *
     * @param destinationAgentName
     *        the name of the agent acting as the destination of the
     *        transfer.
     *
     * @param environmentMetaData
     *        meta data about the environment in which the implementation
     *        of this method is running. This information can only be read,
     *        it cannot be updated by the implementation. The constants
     *        defined in <code>EnvironmentMetaDataConstants</code> class can
     *        be used to access the data held by this map.
     *
     * @param transferMetaData
     *        meta data to associate with the transfer. The information can
     *        only be read, it cannot be updated by the implementation. This
     *        map may also contain keys with IBM reserved names. These
     *        entries are defined in the <code>TransferMetaDataConstants</code>
     *        class and have special semantics.
     *
     * @param fileResults
     *        a list of file transfer result objects that describe the source
     *        file name, destination file name and result of each file transfer
     *        operation attempted.
     *
     * @return
     *        an optional description to enter into the log message describing
     *        transfer completion. A value of <code>null</code> can be used
     *        when no description is required.
     */
    String onSourceTransferEnd(TransferExitResult transferExitResult,
                               String sourceAgentName,
                               String destinationAgentName,
                               Map<String, String>environmentMetaData,
                               Map<String, String>transferMetaData,
                               List<FileTransferResult>fileResults);

}

```

## Tarefas relacionadas

[Customizando o MFT com saídas de usuário](#)

## Referências relacionadas

[“Interface SourceTransferStartExit.java” na página 2235](#)

[“Interface DestinationTransferStartExit.java” na página 2211](#)

[“Interface DestinationTransferEndExit.java” na página 2209](#)

[“Interface MonitorExit.java” na página 2229](#)

[“Interface ProtocolBridgeCredentialExit.java” na página 2230](#)

## ***Interface SourceTransferStartExit.java***

## SourceTransferStartExit.java

```
/*
 * Licensed Materials - Property of IBM
 *
 * "Restricted Materials of IBM"
 *
 * 5724-H72
 *
 * © Copyright IBM Corp. 2008, 2024. All Rights Reserved.
 *
 * US Government Users Restricted Rights - Use, duplication or
 * disclosure restricted by GSA ADP Schedule Contract with
 * IBM Corp.
 */
package com.ibm.wmqfte.exitpoint.api;

import java.util.List;
import java.util.Map;

/**
 * An interface that is implemented by classes that want to be invoked as part of
 * user exit routine processing. This interface defines a method that will be
 * invoked immediately prior to starting a transfer on the agent acting as the
 * source of the transfer.
 */
public interface SourceTransferStartExit {

    /**
     * Invoked immediately prior to starting a transfer on the agent acting as
     * the source of the transfer.
     *
     * @param sourceAgentName
     *     the name of the agent acting as the source of the transfer.
     *     This is the name of the agent that the implementation of this
     *     method will be invoked from.
     *
     * @param destinationAgentName
     *     the name of the agent acting as the destination of the
     *     transfer.
     *
     * @param environmentMetaData
     *     meta data about the environment in which the implementation
     *     of this method is running. This information can only be read,
     *     it cannot be updated by the implementation. The constants
     *     defined in EnvironmentMetaDataConstants class can
     *     be used to access the data held by this map.
     *
     * @param transferMetaData
     *     meta data to associate with the transfer. The meta data passed
     *     to this method can be altered, and the changes to will be
     *     reflected in subsequent exit routine invocations. This map may
     *     also contain keys with IBM reserved names. These entries are
     *     defined in the TransferMetaDataConstants class and
     *     have special semantics.
     *
     * @param fileSpecs
     *     a list of file specifications that govern the file data to
     *     transfer. The implementation of this method can add entries,
     *     remove entries, or modify entries in this list and the changes
     *     will be reflected in the files transferred.
     *
     * @return
     *     a transfer exit result object which is used to determine if the
     *     transfer should proceed, or be cancelled.
     */
    TransferExitResult onSourceTransferStart(String sourceAgentName,
        String destinationAgentName,
        Map<String, String> environmentMetaData,
        Map<String, String> transferMetaData,
        List<SourceFileExitFileSpecification> fileSpecs);
}
```

### Tarefas relacionadas

[Customizando o MFT com saídas de usuário](#)

### Referências relacionadas

[“Classe SourceFileExitFileSpecification.java” na página 2233](#)



[“Interface SourceTransferEndExit.java” na página 2234](#)

[“Interface DestinationTransferStartExit.java” na página 2211](#)

[“Interface DestinationTransferEndExit.java” na página 2209](#)

[“Interface MonitorExit.java” na página 2229](#)

[“Interface ProtocolBridgeCredentialExit.java” na página 2230](#)

## ***Interface TransferExitResult.java***

### **TransferExitResult.java**

```
/*
 * Licensed Materials - Property of IBM
 *
 * "Restricted Materials of IBM"
 *
 * 5724-H72
 *
 * © Copyright IBM Corp. 2008, 2024. All Rights Reserved.
 *
 * US Government Users Restricted Rights - Use, duplication or
 * disclosure restricted by GSA ADP Schedule Contract with
 * IBM Corp.
 */

package com.ibm.wmqfte.exitroutine.api;

/**
 * The result of invoking a transfer exit routine. It is composed of a result
 * code, which determines if the transfer should proceed, and an optional explanatory
 * message. The explanation, if present, is entered into the log message.
 */
public class TransferExitResult {

    private final TransferExitResultCode resultCode;
    private final String explanation;

    /**
     * For convenience, a static "proceed" result with no associated explanation
     * message.
     */
    public static final TransferExitResult PROCEED_RESULT =
        new TransferExitResult(TransferExitResultCode.PROCEED, null);

    /**
     * Constructor. Creates a transfer exit result object with a specified result
     * code and explanation.
     *
     * @param resultCode
     *         The result code to associate with the exit result being created.
     *
     * @param explanation
     *         The explanation to associate with the exit result being created.
     *         A value of <code>null</code> can be specified to indicate no
     *         explanation.
     */
    public TransferExitResult(TransferExitResultCode resultCode, String explanation) {
        this.resultCode = resultCode;
        this.explanation = explanation;
    }

    /**
     * Returns the explanation associated with this transfer exit result.
     *
     * @return
     *         the explanation associated with this exit result.
     */
    public String getExplanation() {
        return explanation;
    }

    /**
     * Returns the result code associated with this transfer exit result.
     *
     * @return
     *         the result code associated with this exit result.
     */
}
```

```
public TransferExitResultCode getResultCode() {
    return resultCode;
}
```

### Tarefas relacionadas

[Customizando o MFT com saídas de usuário](#)

### Referências relacionadas

[“Interface SourceTransferStartExit.java” na página 2235](#)

[“Interface DestinationTransferStartExit.java” na página 2211](#)

[“Interface DestinationTransferEndExit.java” na página 2209](#)

[“Interface MonitorExit.java” na página 2229](#)

[“Interface ProtocolBridgeCredentialExit.java” na página 2230](#)

## Formatos de mensagens para mensagens que podem ser colocadas na fila de comandos do MFT Agent

Esses esquemas XML definem os formatos para mensagens que podem ser colocadas na fila de comando do agente para solicitar que o agente execute uma ação.. A mensagem XML pode ser colocada na fila de comandos do agente usando comandos da linha de comandos ou por um aplicativo.

- [Formato da mensagem de solicitação para transferência de arquivos](#)
- [Formatos de mensagem de solicitação do monitor do MFT](#)
- [Formato de mensagem de solicitação de ping no agente MFT](#)
- [Formato de mensagem de resposta do agente MFT](#)

## Referência do sistema de mensagens da REST API

Informações de referência sobre o messaging REST API.

Para obter mais informações sobre como usar o messaging REST API, consulte [Sistema de mensagens usando o REST API](#)

### REST API recursos

Esta coleção de tópicos fornece informações de referência para cada um desses recursos do messaging REST API

Para obter mais informações sobre como usar o messaging REST API, consulte [Sistema de mensagens usando o REST API](#)

#### ***/messaging/qmgr/{qmgrName}/queue/{queueName}/message***

A API REST do sistema de mensagens permite que as mensagens sejam colocadas em uma fila ou que as mensagens sejam procuradas ou obtenham destrutivamente de uma fila usando o recurso `/messaging/qmgr/{qmgrName}/queue/{queueName}/message`.

#### ***POST /messaging/qmgr/{qmgrName}/queue/{queueName}/message***

É possível usar o método POST HTTP com o recurso `/messaging/qmgr/{qmgrName}/queue/{queueName}/message` para colocar mensagens para a fila especificada no gerenciador de filas especificado

Coloca uma mensagem IBM MQ contendo o corpo da solicitação HTTP no gerenciador de filas e fila especificados. O gerenciador de filas deve estar na mesma máquina que o servidor mqweb. O método suporta apenas corpos de solicitação de HTTP baseados em texto As mensagens são enviadas como

mensagens formatadas MQSTR ou JMS `TextMessage` e são colocadas usando o contexto do usuário atual

**V 9.3.0** A API REST V3 inclui a capacidade de especificar propriedades de mensagem definidas pelo usuário e incluir prioridade de mensagem. Os cabeçalhos de solicitação `ibm-mq-md-priority` e `ibm-mq-usr` estão disponíveis apenas com a REST API V3. O cabeçalho da solicitação `ibm-mq-md-correlationId` tem um formato diferente na API REST V3. O cabeçalho pode ser um ID específico do aplicativo ou, se uma sequência codificada, requer o prefixo `ID:` . Se sua solicitação POST contiver mensagens definidas pelo usuário ou ID de correlação específico do aplicativo, a mensagem será formatada como um JMS `TextMessage`.

- [“URL do recurso” na página 2239](#)
- [“Cabeçalhos de solicitação” na página 2240](#)
- [“Formato do corpo da solicitação:” na página 2242](#)
- [“Requisitos de segurança” na página 2242](#)
- [“Códigos de status de resposta.” na página 2243](#)
- [“Cabeçalhos de resposta” na página 2243](#)
- [“Formato do corpo de resposta” na página 2244](#)
- [“Exemplos” na página 2244](#)

## URL do recurso

`https://host:port/ibmmq/rest/v1/messaging/qmgr/{qmgrName}/queue/{queueName}/message`

`https://host:port/ibmmq/rest/v2/messaging/qmgr/{qmgrName}/queue/{queueName}/message`

**V 9.3.0** `https://host:port/ibmmq/rest/v3/messaging/qmgr/{qmgrName}/queue/{queueName}/message`

## qmgrName

Especifica o nome do gerenciador de filas ao qual se conectar para sistema de mensagens. O gerenciador de filas deve estar na mesma máquina que o servidor mqweb.

**V 9.3.3** A partir do IBM MQ 9.3.3, é possível conectar-se a um gerenciador de filas locais ou a um gerenciador de filas remotas O nome especificado para **qmgrName** depende de como seu servidor mqweb está configurado:

- Se o seu servidor mqweb estiver configurado para se conectar apenas aos gerenciadores de filas locais, use o nome do gerenciador de filas.
- Se o seu servidor mqweb estiver configurado para conectar gerenciadores de filas remotas, use o nome exclusivo para o gerenciador de filas, conforme especificado nas informações de conexão do gerenciador de filas remotas

É possível determinar se seu servidor mqweb está configurado para se conectar a gerenciadores de fila locais ou gerenciadores de filas remotas usando o comando `dspmweb properties` e visualizando a propriedade `mqRestMessagingConnectionMode` .

O nome do gerenciador de filas faz distinção entre maiúsculas e minúsculas.

Se o nome do gerenciador de filas incluir uma barra, um ponto ou um sinal de percentual, esses caracteres deverão ser codificados para URL:

- Uma barra deve ser codificada como `%2F`.
- Um período deve ser codificado como `%2E`.
- Um sinal de percentual deve ser codificado como `%25`.

## queueName

Especifica o nome da fila na qual colocar a mensagem.

A fila deve ser definida como sendo local, remota ou um alias para o gerenciador de filas especificado-ela também pode fazer referência a uma fila em cluster

O nome da fila faz distinção entre maiúsculas e minúsculas..

Se o nome da fila incluir uma barra ou um sinal de porcentagem, esses caracteres deverão ser codificados em URL :

- Uma barra, /, deve ser codificada como %2F.
- Um sinal de porcentagem, %, deve ser codificado como %25.

Será possível usar HTTP (Protocolo de Transporte de Hipertexto) no lugar de HTTPS se você ativar as conexões HTTP. Para obter mais informações sobre como ativar HTTP, veja [Configurando as portas HTTP e HTTPS](#).

## Cabeçalhos de solicitação

Os cabeçalhos a seguir devem ser enviados com a solicitação:

### Autorização

Esse cabeçalho deverá ser enviado se você estiver usando a autenticação básica. Para obter mais informações, veja [Usando a autenticação básica HTTP com a REST API](#).

### Content-Type

Esse cabeçalho deve ser enviado com um dos seguintes valores:

- text/plain;charset=utf-8
- text/html;charset=utf-8
- text/xml;charset=utf-8
- application/json;charset=utf-8
- application/xml;charset=utf-8

### ibm-mq-rest-csrf-token

Esse cabeçalho deve ser configurado, mas o valor pode ser qualquer coisa, incluindo ficar em branco.

Os cabeçalhos a seguir podem ser enviados opcionalmente com a solicitação:

### Accept-Language

Este cabeçalho especifica o idioma necessário para quaisquer exceções ou mensagens de erro retornados no corpo da mensagem de resposta

### **ibm-mq-md-correlationId**

Esse cabeçalho configura o ID de correlação da mensagem criada. O cabeçalho é especificado como uma sequência codificada hexadecimal de 48 caracteres, representando 24 bytes.. Não prefixe o valor com "ID: ", a API REST inclui essa sequência automaticamente.

Por exemplo: `ibm-mq-md-correlationId: 414d5120514d4144455620202020202067d8bf5923582e02`

### **ibm-mq-md-correlationId**

Esse cabeçalho configura o ID de correlação da mensagem criada. O ID de correlação pode ter um dos seguintes formatos:

- Uma sequência codificada hexadecimal de 48 caracteres, representando 24 bytes, prefixada com a sequência "ID: ".. Por exemplo: `ibm-mq-md-correlationId: ID: 414d5120514d4144455620202020202067d8bf5923582e02`
- Um valor específico do aplicativo. O valor é uma sequência específica do aplicativo: `ibm-mq-md-correlationId: My-Custom-CorrelId`

Se você especificar esse formato de ID de correlação, o destino da mensagem será destinado como WMQ\_CLIENT\_JMS\_COMPLIANT e, portanto, incorporará um cabeçalho MQRFH2 .

### **ibm-mq-md-expiração**

Este cabeçalho configura a duração de expiração para a mensagem criada. A expiração de uma mensagem é iniciada a partir do momento em que a mensagem chega na fila. Como resultado, a latência de rede é ignorada. O cabeçalho deve ser especificado como um dos seguintes valores:

#### **ilimitado**

A mensagem não expira.

Esse valor é o valor padrão.

#### **Valor de número inteiro**

Milissegundos antes da expiração da mensagem.

Limitado ao intervalo de 0 a 99999999900.

### **ibm-mq-md-persistence**

Esse cabeçalho configura a persistência para a mensagem criada. O cabeçalho é especificado como um dos seguintes valores:

#### **nonPersistent**

A mensagem não sobrevive a falhas do sistema ou reinícios do gerenciador de filas.

Esse valor é o valor padrão.

#### **persistente**

A mensagem sobrevive a falhas do sistema ou reinícios do gerenciador de filas.

V 9.3.0

REST API V3

### **ibm-mq-md-priority**

Para a API de REST V3, esse cabeçalho configura a prioridade da mensagem criada. O cabeçalho é especificado como um dos seguintes valores:

#### **asDestination**

A mensagem usa a prioridade especificada no atributo DEFPRTY do objeto de fila IBM MQ subjacente.

Esse valor é o valor padrão.

#### **Valor de número inteiro**

Especifique a prioridade real como um número inteiro no intervalo de 0 a 9.

Por exemplo: `ibm-mq-md-priority: asDestination`

Para a API REST V1 e a API REST V2, a prioridade da mensagem para POST é sempre 4.

### **ibm-mq-md-replyTo**

Este cabeçalho configura o destino de resposta para a mensagem criada. O formato do cabeçalho usa a notação padrão de fornecer a fila de respostas e um gerenciador de filas opcional:

`replyQueue[@replyQmgr]`

Por exemplo: `ibm-mq-md-replyTo: myReplyQueue@myReplyQMGR`

V 9.3.0

REST API V3

### **ibm-mq-usr**

Esse cabeçalho configura as propriedades definidas pelo usuário da mensagem de solicitação.

As propriedades têm a seguinte sintaxe:

`ibm-mq-usr: property_name; user_value; user_type`

#### **property\_name**

O nome da propriedade do usuário que está sendo especificada. Este deve ser um nome de propriedade JMS válido.

#### **user\_value**

O valor da propriedade.

#### **user\_type**

O tipo da propriedade:

- `boolean` (true/false, MQBOOL)
- `byte` (número inteiro de 8 bits, MQINT8)

- `short` (número inteiro de 16 bits, MQINT16)
- `integer` (número inteiro de 32 bits, MQINT32)
- `long` (número inteiro de 64 bits, MQINT64)
- `float` (32 bits reais, MQFLOAT32)
- `double` (real de 64 bits, MQFLOAT64)
- `string` (sequência entre aspas)

É possível configurar várias propriedades em uma mensagem. É possível especificar várias propriedades separadas por vírgula em um único cabeçalho de solicitação `ibm-mq-usr` ou usar duas ou mais instâncias separadas do cabeçalho de solicitação `ibm-mq-usr`.

Por exemplo, é possível configurar múltiplas propriedades definidas pelo usuário em um único cabeçalho: `ibm-mq-usr: userPropertyA;5;byte,userPropertyB;-10;integer`.

Ou você pode configurar várias propriedades definidas pelo usuário em instâncias separadas do cabeçalho: `ibm-mq-usr: userPropertyA;5;byte` `ibm-mq-usr: userPropertyB;-10;integer`

## Formato do corpo da solicitação:

O corpo da solicitação deve ser de texto e usar a codificação UTF-8. Não é necessária uma estrutura de texto específica. Uma mensagem formatada MQSTR contendo o texto do corpo da solicitação é criada e colocada na fila especificada..

**V 9.3.0** **REST API V3** Se as propriedades definidas pelo usuário da API REST V3 ou os recursos do ID de correlação específicos do aplicativo forem usados, uma mensagem formatada JMS `TextMessage` contendo o texto do corpo da solicitação será criada e colocada na fila especificada.

Para obter mais informações, consulte [exemplos](#).

## Requisitos de segurança

O responsável pela chamada deve ser autenticado no servidor mqweb. As funções `MQWebAdmin` e `MQWebAdminRO` não são aplicáveis à messaging REST API. Para obter mais informações sobre segurança para o REST API, consulte [Segurança do IBM MQ Console e do REST API](#).

Depois de autenticado para o servidor mqWeb, o usuário é capaz de usar o messaging REST API e o administrative REST API.

O principal de segurança do responsável pela chamada deve ter a capacidade de colocar mensagens na fila especificada:

- A fila especificada pela parte `{queueName}` da URL do recurso precisa ser PUT ativada.
- **MQ Appliance** **ALW** Para a fila especificada pela parte `{queueName}` do recurso URL, a autoridade +PUT deve ser concedida ao principal de segurança do responsável pela chamada.
- **z/OS** Para a fila especificada pela parte `{queueName}` do recurso URL, UPDATE o acesso deve ser concedido ao proprietário de segurança do responsável pela chamada.

Esse proprietário de segurança pode ser o usuário que iniciou o servidor mqweb, ou o usuário que efetuou login no servidor mqweb. Se o gerenciador de filas ao qual você se conectar for um gerenciador de filas remotas, o principal de segurança poderá, em vez disso, ser o usuário fornecido pelas credenciais nas informações de conexão do gerenciador de filas remotas ou outro usuário determinado pelas regras de segurança do canal. Para obter mais informações, consulte [Determinando o principal de segurança usado pelo messaging REST API](#).

**ALW** No AIX, Linux, and Windows, é possível conceder autoridade às entidades de segurança para usar os recursos do IBM MQ usando o comando `setmqaut`. Para obter mais informações, consulte `setmqaut` (autoridade de concessão ou revogação).

Se você usar Advanced Message Security (AMS) com o messaging REST API, observe que todas as mensagens serão criptografadas usando o contexto do servidor mqweb, não o contexto do usuário que postar a mensagem.

## Códigos de status de resposta.

### 201

Mensagem criada e enviada com êxito.

### 400

Dados inválidos fornecidos.

Por exemplo, um valor de cabeçalho de solicitação inválido foi especificado.

### 401

Não autenticado.

O ouvinte precisa ser autenticado para o servidor mqweb e ser um membro de um ou mais das funções MQWebAdmin, MQWebAdminRO ou MQWebUser. O cabeçalho `ibm-mq-rest-csrf-token` também deve ser especificado.. Para obter mais informações, consulte [“Requisitos de segurança”](#) na página 2242.

### 403

Não autorizado.

O responsável pela chamadas é autenticado para o servidor mqweb e está associado a um proprietário válido No entanto, o proprietário não tem acesso a todos ou a um subconjunto dos recursos necessários do IBM MQ ou não está na função MQWebUser .. Para obter mais informações sobre o acesso necessário, consulte [“Requisitos de segurança”](#) na página 2242.

### 404

A fila não existe.

### 405

A fila é inibida por PUT

### 415

Um cabeçalho ou corpo da mensagem é um tipo de mídia não suportado..

Por exemplo, o cabeçalho `Content-Type` é configurado para um tipo de mídia não suportado

### 500

Problema do servidor ou código de erro de IBM MQ.

### 502

O principal de segurança atual não pode enviar a mensagem, pois o provedor de sistemas de mensagens não suporta a função necessária Por exemplo, se o caminho da classe do servidor mqweb for inválido,

### 503

Gerenciador de filas não em execução.

## Cabeçalhos de resposta

Os seguintes cabeçalhos são retornados com a resposta:

### Conteúdo-Idioma

Especifica o identificador de idioma da mensagem de resposta no caso de quaisquer erros ou exceções. Usado em conjunto com o cabeçalho da solicitação `Accept-Language` para indicar o idioma necessário para quaisquer condições de erro ou de exceção O padrão do servidor mqweb será usado se o idioma solicitado não for suportado

### Content-Length

Especifica o comprimento do corpo de resposta HTTP , mesmo quando não há conteúdo.. Após o sucesso, o valor é zero.

## Content-Type

Especifica o tipo de corpo de resposta Após o sucesso, o valor é `text/plain; charset=utf-8`. No caso de erros ou exceções, o valor será `application/json; charset=utf-8`.

### REST API V1 > REST API V2 **ibm-mq-md-messageId**

Especifica o ID de mensagem que é alocado por IBM MQ para esta mensagem.. Como o cabeçalho da solicitação `ibm-mq-md-correlationId`, ele é representado como uma sequência codificada hexadecimal de 48 caracteres, representando 24 bytes..

Por exemplo:

```
ibm-mq-md-messageId: 414d5120514d4144455620202020202067d8ce5923582f07
```

### V 9.3.0 > REST API V3 **ibm-mq-md-messageId**

Especifica o ID de mensagem que é alocado por IBM MQ para esta mensagem.. Assim como o cabeçalho da solicitação `ibm-mq-md-correlationId`, ele é representado como uma sequência codificada hexadecimal de 48 caracteres, representando 24 bytes, prefixados pela sequência ID:

Por exemplo:

```
ibm-mq-md-messageId: ID:414d5120514d4144455620202020202067d8ce5923582f07
```

### V 9.3.3 **ibm-mq-resolved-qmgr**

Especifica o nome do gerenciador de filas que foi usado para a solicitação Se um nome exclusivo foi usado na URL do recurso, esse cabeçalho identifica o nome do gerenciador de filas que está associado a esse nome exclusivo. Se o nome exclusivo usado no recurso URL se referir a um grupo de gerenciadores de filas, esse cabeçalho identificará qual gerenciador de filas dentro do grupo foi usado.

## Formato do corpo de resposta

O corpo de resposta estará vazio se a mensagem for enviada com êxito Se ocorrer um erro, o corpo de resposta conterá uma mensagem de erro Para obter mais informações, consulte [REST API manipulação de erros](#).

## Exemplos

Os exemplos a seguir usam a URL de recurso v2. Se você estiver usando uma versão do IBM MQ anterior ao IBM MQ 9.1.5, use a URL do recurso v1 como alternativa. Ou seja, na URL do recurso, substitua v1 em que a URL de exemplo usa v2.

O exemplo a seguir registra um usuário chamado `mquser` com a senha `mquser`. Em cURL, a solicitação de login pode ser semelhante à seguinte Windows exemplo. O token LTPA é armazenado no arquivo `cookiejar.txt` usando a sinalização `-c`:

```
curl -k "https://localhost:9443/ibmmq/rest/v2/login" -X POST
-H "Content-Type: application/json" --data "{\"username\": \"mquser\", \"password\": \"mquser\"}"
-c c:\cookiejar.txt
```

Após o usuário efetuar login, o token LTPA e o cabeçalho `ibm-mq-rest-csrf-token` HTTP são usados para autenticar solicitações adicionais. O `ibm-mq-rest-csrf-token token_value` pode ser qualquer valor, incluindo em branco.

- O exemplo Windows cURL a seguir envia uma mensagem para a fila Q1 no gerenciador de filas QM1, usando opções padrão. A mensagem contém o texto `"Hello World!"`:

```
curl -k "https://localhost:9443/ibmmq/rest/v2/messaging/qmgr/QM1/queue/Q1/message"
-X POST -b c:\cookiejar.txt -H "ibm-mq-rest-csrf-token: token_value"
-H "Content-Type: text/plain; charset=utf-8" --data "Hello World!"
```



- O exemplo Windows cURL a seguir envia uma mensagem persistente para a fila Q1 no gerenciador de filas QM1, com uma expiração de 2 minutos. A mensagem contém o texto "Hello World!":

```
curl -k "https://localhost:9443/ibmmq/rest/v2/messaging/qmgr/QM1/queue/Q1/message"
-X POST -b c:\cookiejar.txt -H "ibm-mq-rest-csrf-token: token_value"
-H "Content-Type: text/plain;charset=utf-8" -H "ibm-mq-md-persistence: persistent"
-H "ibm-mq-md-expiry: 120000" --data "Hello World!"
```

- O exemplo Windows cURL a seguir envia uma mensagem não persistente para a fila Q1 no gerenciador de filas QM1, sem ID de correlação definido e de expiração. A mensagem contém o texto "Hello World!":

```
curl -k "https://localhost:9443/ibmmq/rest/v2/messaging/qmgr/QM1/queue/Q1/message"
-X POST -b c:\cookiejar.txt -H "ibm-mq-rest-csrf-token: token_value"
-H "Content-Type: text/plain;charset=utf-8" -H "ibm-mq-md-persistence: nonPersistent"
-H "ibm-mq-md-expiry: unlimited" -H "ibm-mq-md-correlationId:
414d5120514d4144455620202020202067d8b
f5923582e02" --data "Hello World!"
```

### **GET /messaging/qmgr/{qmgrName}/queue/{queueName}/message**

É possível usar o método GET HTTP com o recurso /messaging/qmgr/{qmgrName}/queue/{queueName}/message para procurar mensagens do gerenciador de filas e da fila associados.

Procura a primeira mensagem disponível do gerenciador de filas e da fila especificados. O gerenciador de filas deve estar na mesma máquina que o servidor mqweb. O corpo da mensagem é retornado na resposta HTTP . A mensagem deve ter um formato de MQSTR ou JMS TextMessage e é recebido usando o contexto do usuário atual

Todas as mensagens são deixadas na fila e um código de status apropriado é retornado para o responsável pela chamada para quaisquer mensagens inapropriadas Por exemplo, uma mensagem que não possui um formato MQSTR ou JMS TextMessage

**V9.3.0** A API REST V3 inclui a capacidade de especificar propriedades de mensagem definidas pelo usuário e incluir prioridade de mensagem com mensagens. Os cabeçalhos de resposta ibm-mq-md-priority e ibm-mq-usr estão disponíveis apenas com a API REST V3. O cabeçalho da solicitação ibm-mq-md-correlationId tem um formato diferente na API REST V3. O cabeçalho pode ser um ID específico do aplicativo ou, se uma sequência codificada, retém o prefixo ID: .. O cabeçalho de resposta ibm-mq-md-messageId e o parâmetro de consulta têm um formato diferente na API REST V3, ele retém o prefixo ID: .

- [“URL do recurso” na página 2245](#)
- [“Parâmetros de consulta opcionais” na página 2246](#)
- [“Cabeçalhos de solicitação” na página 2247](#)
- [“Formato do corpo da solicitação:” na página 2247](#)
- [“Requisitos de segurança” na página 2247](#)
- [“Códigos de status de resposta.” na página 2248](#)
- [“Cabeçalhos de resposta” na página 2249](#)
- [“Formato do corpo de resposta” na página 2251](#)
- [“Exemplos” na página 2251](#)

### **URL do recurso**

https://host:port/ibmmq/rest/v1/messaging/qmgr/{qmgrName}/queue/{queueName}/message

https://host:port/ibmmq/rest/v2/messaging/qmgr/{qmgrName}/queue/{queueName}/message

**V9.3.0** https://host:port/ibmmq/rest/v3/messaging/qmgr/{qmgrName}/queue/{queueName}/message

## qmgrName

Especifica o nome do gerenciador de filas ao qual se conectar para sistema de mensagens. O gerenciador de filas deve estar na mesma máquina que o servidor mqweb.

**V 9.3.3** A partir do IBM MQ 9.3.3, é possível conectar-se a um gerenciador de filas locais ou a um gerenciador de filas remotas O nome especificado para **qmgrName** depende de como seu servidor mqweb está configurado:

- Se o seu servidor mqweb estiver configurado para se conectar apenas aos gerenciadores de filas locais, use o nome do gerenciador de filas.
- Se o seu servidor mqweb estiver configurado para conectar gerenciadores de filas remotas, use o nome exclusivo para o gerenciador de filas, conforme especificado nas informações de conexão do gerenciador de filas remotas

É possível determinar se seu servidor mqweb está configurado para se conectar a gerenciadores de fila locais ou gerenciadores de filas remotas usando o comando **dspmweb properties** e visualizando a propriedade **mqRestMessagingConnectionMode** .

O nome do gerenciador de filas faz distinção entre maiúsculas e minúsculas.

Se o nome do gerenciador de filas incluir uma barra, um ponto ou um sinal de percentual, esses caracteres deverão ser codificados para URL:

- Uma barra (/) precisa ser codificada como %2F.
- Um sinal de percentual (%) precisa ser codificado como %25.

## queueName

Especifica o nome da fila a partir da qual procurar a mensagem.

A fila deve estar definida como sendo local ou um alias que aponte para uma fila local.

O nome da fila faz distinção entre maiúsculas e minúsculas..

Se o nome da fila incluir uma barra ou um sinal de porcentagem, esses caracteres deverão ser codificados em URL :

- Uma barra, /, deve ser codificada como %2F.
- Um sinal de porcentagem, %, deve ser codificado como %25.

Será possível usar HTTP (Protocolo de Transporte de Hipertexto) no lugar de HTTPS se você ativar as conexões HTTP. Para obter mais informações sobre como ativar HTTP, veja [Configurando as portas HTTP e HTTPS](#).

## Parâmetros de consulta opcionais

**REST API V1** **REST API V2** **correlationId=hexValue**

Especifica que o método HTTP retorna a próxima mensagem com o ID de correlação correspondente.

### hexValue

O parâmetro de consulta deve ser especificado como uma cadeia codificada hexadecimal de 48 caracteres, representando 24 bytes..

Por exemplo:

```
../message?correlationId=414d5120514d4144455620202020202067d8bf5923582e02
```

**V 9.3.0** **REST API V3** **correlationId= ID:hexValue ou correlationId=application\_specific\_value**

Especifica que o método HTTP retorna a próxima mensagem com o ID de correlação correspondente.

### hexValue

O parâmetro de consulta deve ser especificado como uma sequência codificada hexadecimal de 48 caracteres, representando 24 bytes, e precedido pela sequência " ID: "

Por exemplo:

```
../message?correlationId=ID:414d5120514d4144455620202020202067d8bf5923582e02
```

### **application\_specific\_value**

O parâmetro de consulta pode ser especificado como uma sequência específica do aplicativo

Por exemplo:

```
../message?correlationId=My-Custom-CorrelId
```

### **REST API V1** **REST API V2** **messageId=hexValue**

Especifica que o método HTTP retorna a próxima mensagem com o ID de mensagem correspondente.

#### **hexValue**

O parâmetro de consulta deve ser especificado como uma cadeia codificada hexadecimal de 48 caracteres, representando 24 bytes..

Por exemplo:

```
../message?messageId=414d5120514d4144455620202020202067d8ce5923582f07
```

### **V 9.3.0** **REST API V3** **messageId= ID:hexValue**

Especifica que o método HTTP retorna a próxima mensagem com o ID de mensagem correspondente.

#### **hexValue**

O parâmetro de consulta deve ser especificado como uma sequência codificada hexadecimal de 48 caracteres, representando 24 bytes, e precedido pela sequência " ID: "

Por exemplo:

```
../message?messageId=ID:414d5120514d4144455620202020202067d8ce5923582f07
```

## **Cabeçalhos de solicitação**

Os cabeçalhos a seguir devem ser enviados com a solicitação:

### **Autorização**

Esse cabeçalho deverá ser enviado se você estiver usando a autenticação básica. Para obter mais informações, veja [Usando a autenticação básica HTTP com a REST API](#).

### **ibm-mq-rest-csrf-token**

Esse cabeçalho deve ser configurado, mas o valor pode ser qualquer coisa, incluindo ficar em branco.

Os cabeçalhos a seguir podem ser enviados opcionalmente com a solicitação:

### **Aceitar-Conjunto de caracteres**

Esse cabeçalho pode ser usado para indicar qual conjunto de caracteres é aceitável para a resposta. Se for especificado, esse cabeçalho deverá ser configurado como UTF-8.

### **Accept-Language**

Este cabeçalho especifica o idioma necessário para quaisquer exceções ou mensagens de erro retornados no corpo da mensagem de resposta.

## **Formato do corpo da solicitação:**




Nenhum.

## **Requisitos de segurança**


O responsável pela chamada deve ser autenticado no servidor mqweb. As funções MQWebAdmin e MQWebAdminRO não são aplicáveis à messaging REST API. Para obter mais informações sobre segurança para o REST API, consulte [Segurança do IBM MQ Console e do REST API](#).

Depois de autenticado para o servidor mqWeb, o usuário é capaz de usar o messaging REST API e o administrative REST API.

A capacidade de procurar mensagens da fila especificada deve ser concedida à entidade de segurança do responsável pela chamada:

- A fila especificada pela parte *{queueName}* da URL do recurso precisa ser BROWSE ativada.
-   Para a fila especificada pela parte *{queueName}* da URL do recurso, +GET, +INQ e a autoridade do +BROWSE autoridade precisa ser concedida à entidade de segurança do responsável pela chamada.
-  Para a fila especificada pela parte *{queueName}* da URL do recurso, UPDATE, o acesso precisa ser concedido à entidade de segurança do responsável pela chamada.

Esse proprietário de segurança pode ser o usuário que iniciou o servidor mqweb, ou o usuário que efetuou login no servidor mqweb. Se o gerenciador de filas ao qual você se conectar for um gerenciador de filas remotas, o principal de segurança poderá, em vez disso, ser o usuário fornecido pelas credenciais nas informações de conexão do gerenciador de filas remotas ou outro usuário determinado pelas regras de segurança do canal. Para obter mais informações, consulte [Determinando o principal de segurança usado pelo messaging REST API](#).

 No AIX, Linux, and Windows, é possível conceder autoridade às entidades de segurança para usar os recursos do IBM MQ usando o comando **setmqaut**. Para obter mais informações, consulte [setmqaut](#) (autoridade de concessão ou revogação).

 No z/OS, consulte [Configurando a segurança no z/OS](#).

## Códigos de status de resposta.

### 200

Mensagem recebida com êxito.

### 204

Não há mensagens disponíveis.

### 400

Dados inválidos fornecidos.

Por exemplo, foi especificado um valor de parâmetro de consulta inválido.

### 401

Não autenticado.

O ouvinte precisa ser autenticado para o servidor mqweb e ser um membro de um ou mais das funções MQWebAdmin, MQWebAdminRO ou MQWebUser. O cabeçalho `ibm-mq-rest-csrf-token` também deve ser especificado.. Para obter mais informações, consulte [“Requisitos de segurança” na página 2247](#).

### 403

Não autorizado.

O responsável pela chamadas é autenticado para o servidor mqweb e está associado a um proprietário válido No entanto, o proprietário não tem acesso a todos ou a um subconjunto dos recursos necessários do IBM MQ ou não está na função MQWebUser .. Para obter mais informações sobre o acesso necessário, consulte [“Requisitos de segurança” na página 2247](#).

### 404

A fila não existe.

### 500

Problema do servidor ou código de erro de IBM MQ.

### 501

A resposta de HTTP não pôde ser construída

Por exemplo, a mensagem recebida tem um tipo incorreto ou tem o tipo correto, mas o corpo não pôde ser processado

## 502

O principal de segurança atual não pode receber a mensagem porque o provedor de sistemas de mensagens não suporta a função necessária. Por exemplo, se o caminho da classe do servidor mqweb for inválido,

## 503

Gerenciador de filas não em execução.

## Cabeçalhos de resposta

Os seguintes cabeçalhos são retornados com a resposta:

### Conteúdo-Idioma

Especifica o identificador de idioma da mensagem de resposta no caso de quaisquer erros ou exceções. Usado em conjunto com o cabeçalho da solicitação `Accept-Language` para indicar o idioma necessário para quaisquer condições de erro ou de exceção. O padrão do servidor mqweb será usado se o idioma solicitado não for suportado.

### Content-Length

Especifica o comprimento do corpo de resposta HTTP, mesmo quando não há conteúdo. O valor contém o comprimento (bytes) dos dados da mensagem.

### Content-Type

Especifica o tipo de conteúdo retornado no corpo de resposta da mensagem recebida. Após o sucesso, o valor é `text/plain; charset=utf-8`. No caso de erros ou exceções, o valor será `application/json; charset=utf-8`.

### REST API V1 REST API V2 **ibm-mq-md-correlationId**

Especifica o ID de correlação da mensagem recebida. O cabeçalho será retornado se a mensagem recebida contiver um ID de correlação válido. É representado como uma sequência codificada hexadecimal de 48 caracteres, representando 24 bytes.

Por exemplo:

```
ibm-mq-md-correlationId: 414d5120514d4144455620202020202067d8bf5923582e02
```

### V 9.3.0 REST API V3 **ibm-mq-md-correlationId**

Especifica o ID de correlação da mensagem recebida. O cabeçalho será retornado se a mensagem recebida contiver um ID de correlação válido. O ID de correlação pode ter um dos seguintes formatos:

- Uma sequência codificada hexadecimal de 48 caracteres, representando 24 bytes, prefixada com a sequência "ID: ". Por exemplo:

```
ibm-mq-md-correlationId: ID:414d5120514d4144455620202020202067d8bf5923582e02
```

- Um valor específico do aplicativo. O valor é uma sequência específica do aplicativo:

```
ibm-mq-md-correlationId: My-Custom-CorrelId
```

### ibm-mq-md-expiração

Especifica a duração de expiração restante de uma mensagem recebida. O cabeçalho pode ser um dos seguintes valores:

#### ilimitado

A mensagem não expira.

#### Valor de número inteiro

Os milissegundos restantes antes da expiração da mensagem

### REST API V1 REST API V2 **ibm-mq-md-messageId**

Especifica o ID de mensagem que é alocado por IBM MQ para esta mensagem.. Como o cabeçalho `ibm-mq-md-correlationId`, ele é representado como uma sequência codificada hexadecimal de 48 caracteres, representando 24 bytes..

Por exemplo:

```
ibm-mq-md-messageId: 414d5120514d4144455620202020202067d8ce5923582f07
```

### V 9.3.0 REST API V3 **ibm-mq-md-messageId**

Especifica o ID de mensagem que é alocado por IBM MQ para esta mensagem.. Como o cabeçalho `ibm-mq-md-correlationId`, ele é representado como uma sequência codificada hexadecimal de 48 caracteres, representando 24 bytes, prefixados com a sequência "ID: "

Por exemplo:

```
ibm-mq-md-messageId: ID:414d5120514d4144455620202020202067d8ce5923582f07
```

### **ibm-mq-md-persistence**

Especifica a persistência da mensagem recebida.. O cabeçalho pode ser um dos seguintes valores:

#### **nonPersistent**

A mensagem não sobrevive a falhas do sistema ou reinícios do gerenciador de filas.

#### **persistente**

A mensagem sobrevive a falhas do sistema ou reinícios do gerenciador de filas.

### V 9.3.0 REST API V3 **ibm-mq-md-priority**

Retorna a configuração da prioridade da mensagem Por exemplo:

```
ibm-mq-md-priority: 3
```

### **ibm-mq-md-replyTo**

Especifica o destino de resposta para a mensagem recebida O formato do cabeçalho usa a notação padrão da fila de resposta e do gerenciador de fila, `replyQueue@replyQMgr`

Por exemplo:

```
ibm-mq-md-replyTo: myReplyQueue@myReplyQMgr
```

### V 9.3.3 **ibm-mq-resolved-qmgr**

Especifica o nome do gerenciador de filas que foi usado para a solicitação Se um nome exclusivo foi usado na URL do recurso, esse cabeçalho identifica o nome do gerenciador de filas que está associado a esse nome exclusivo. Se o nome exclusivo usado no recurso URL se referir a um grupo de gerenciadores de filas, esse cabeçalho identificará qual gerenciador de filas dentro do grupo foi usado.

### V 9.3.0 REST API V3 **ibm-mq-usr**

Retorna as propriedades definidas pelo usuário da mensagem Diversas propriedades podem ser configuradas em uma mensagem, nesse caso, haverá duas ou mais instâncias separadas do cabeçalho de resposta `ibm-mq-usr`

Por exemplo:

```
ibm-mq-usr: myIPprop;5;short
ibm-mq-usr: mySProp;"hi";string
ibm-mq-usr: myBProp>true;boolean
```

As propriedades têm a seguinte sintaxe:

```
ibm-mq-usr: property_name; user_value; user_type
```

**property\_name**

O nome da propriedade do usuário que está sendo especificada. Este deve ser um nome de propriedade JMS válido.

**user\_value**

O valor da propriedade.

**user\_type**

O tipo da propriedade:

- `boolean` (true/false, MQBOOL)
- `byte` (número inteiro de 8 bits, MQINT8)
- `short` (número inteiro de 16 bits, MQINT16)
- `integer` (número inteiro de 32 bits, MQINT32)
- `long` (número inteiro de 64 bits, MQINT64)
- `float` (32 bits reais, MQFLOAT32)
- `double` (real de 64 bits, MQFLOAT64)
- `string` (sequência entre aspas)

## Formato do corpo de resposta

Após o sucesso, o corpo de resposta contém o corpo da mensagem da mensagem recebida. Se ocorrer um erro, o corpo de resposta conterá uma mensagem de erro formatada por JSON. Ambas as respostas são codificadas por UTF-8. Para obter mais informações, consulte [REST API manipulação de erros](#).

Esteja ciente de que ao receber uma mensagem, apenas mensagens formatadas IBM MQ MQSTR ou JMS TextMessage são suportadas.

Procurar uma fila que foi marcada como GET inibida não retorna nenhum conteúdo.

Se a fila que está sendo procurada contiver mensagens com identificadores de mensagem duplicados, a primeira mensagem será retornada ao filtrar o identificador de mensagem...

## Exemplos

Os exemplos a seguir usam a URL de recurso v2. Se você estiver usando uma versão do IBM MQ anterior ao IBM MQ 9.1.5, use a URL do recurso v1 como alternativa. Ou seja, na URL do recurso, substitua v1 em que a URL de exemplo usa v2.

O exemplo a seguir registra um usuário chamado `muser` com a senha `muser`. Em cURL, a solicitação de login pode ser semelhante à seguinte Windows exemplo. O token LTPA é armazenado no arquivo `cookiejar.txt` usando a sinalização `-c`:

```
curl -k "https://localhost:9443/ibmmq/rest/v2/login" -X POST
-H "Content-Type: application/json" --data "{\"username\":\"muser\", \"password\":\"muser\"}"
-c c:\cookiejar.txt
```

Após o usuário efetuar login, o token LTPA e o cabeçalho `ibm-mq-rest-csrf-token` HTTP são usados para autenticar solicitações adicionais. O `ibm-mq-rest-csrf-token` `token_value` pode ser qualquer valor, incluindo em branco.

- O exemplo de Windows cURL a seguir procura a próxima mensagem disponível da fila Q1 no gerenciador de filas QM1, usando as opções padrão:

```
curl -k "https://localhost:9443/ibmmq/rest/v2/messaging/qmgr/QM1/queue/Q1/message"
-X GET -b c:\cookiejar.txt -H "ibm-mq-rest-csrf-token: token-value"
-H "Accept: text/plain"
```





**V 9.3.0** `https://host:port/ibmmq/rest/v3/messaging/qmgr/{qmgrName}/queue/{queueName}/message`

### qmgrName

Especifica o nome do gerenciador de filas ao qual se conectar para sistema de mensagens. O gerenciador de filas deve estar na mesma máquina que o servidor mqweb.

**V 9.3.3** A partir do IBM MQ 9.3.3, é possível conectar-se a um gerenciador de filas locais ou a um gerenciador de filas remotas O nome especificado para **qmgrName** depende de como seu servidor mqweb está configurado:

- Se o seu servidor mqweb estiver configurado para se conectar apenas aos gerenciadores de filas locais, use o nome do gerenciador de filas.
- Se o seu servidor mqweb estiver configurado para conectar gerenciadores de filas remotas, use o nome exclusivo para o gerenciador de filas, conforme especificado nas informações de conexão do gerenciador de filas remotas

É possível determinar se seu servidor mqweb está configurado para se conectar a gerenciadores de fila locais ou gerenciadores de filas remotas usando o comando **dspmweb properties** e visualizando a propriedade **mqRestMessagingConnectionMode** .

O nome do gerenciador de filas faz distinção entre maiúsculas e minúsculas.

Se o nome do gerenciador de filas incluir uma barra, um ponto ou um sinal de percentual, esses caracteres deverão ser codificados para URL:

- Uma barra (/) precisa ser codificada como %2F.
- Um sinal de percentual (%) precisa ser codificado como %25.

### queueName

Especifica o nome da fila da qual obter a próxima mensagem.

A fila deve ser definida como sendo local ou um alias apontando para uma fila local.

O nome da fila faz distinção entre maiúsculas e minúsculas..

Se o nome da fila incluir uma barra ou um sinal de porcentagem, esses caracteres deverão ser codificados em URL :

- Uma barra, /, deve ser codificada como %2F.
- Um sinal de porcentagem, %, deve ser codificado como %25.

Será possível usar HTTP (Protocolo de Transporte de Hipertexto) no lugar de HTTPS se você ativar as conexões HTTP. Para obter mais informações sobre como ativar HTTP, veja [Configurando as portas HTTP e HTTPS](#).

## Parâmetros de consulta opcionais

**REST API V1** **REST API V2** **correlationId=hexValue**

Especifica que o método HTTP retorna a próxima mensagem com o ID de correlação correspondente.

### hexValue

O parâmetro de consulta deve ser especificado como uma cadeia codificada hexadecimal de 48 caracteres, representando 24 bytes..

Por exemplo:

```
../message?correlationId=414d5120514d4144455620202020202067d8bf5923582e02
```

**V 9.3.0** **REST API V3** **correlationId= ID:hexValue ou correlationId=application\_specific\_value**

Especifica que o método HTTP retorna a próxima mensagem com o ID de correlação correspondente.

### hexValue

O parâmetro de consulta deve ser especificado como uma sequência codificada hexadecimal de 48 caracteres, representando 24 bytes, e precedido pela sequência "ID: "

Por exemplo:

```
../message?correlationId=ID:414d5120514d4144455620202020202067d8bf5923582e02
```

### application\_specific\_value

O parâmetro de consulta pode ser especificado como uma sequência específica do aplicativo

Por exemplo:

```
../message?correlationId=My-Custom-CorrelId
```

### REST API V1 → REST API V2 **messageId=hexValue**

Especifica que o método HTTP retorna a próxima mensagem com o ID de mensagem correspondente.

### hexValue

O parâmetro de consulta deve ser especificado como uma cadeia codificada hexadecimal de 48 caracteres, representando 24 bytes..

Por exemplo:

```
../message?messageId=414d5120514d4144455620202020202067d8ce5923582f07
```

### V 9.3.0 → REST API V3 **messageId= ID:hexValue**

Especifica que o método HTTP retorna a próxima mensagem com o ID de mensagem correspondente.

### hexValue

O parâmetro de consulta deve ser especificado como uma sequência codificada hexadecimal de 48 caracteres, representando 24 bytes, e precedido pela sequência "ID: "

Por exemplo:

```
../message?messageId=ID:414d5120514d4144455620202020202067d8ce5923582f07
```

### **wait=integerValue**

Especifica que o método HTTP aguardará *integerValue* milissegundos para que a próxima mensagem se torne disponível.

### integerValue

O parâmetro de consulta deve ser especificado como um valor inteiro que representa a duração de milissegundos. O valor máximo é 2147483647.

Por exemplo:

```
../message?wait=120000
```

## Cabeçalhos de solicitação

Os cabeçalhos a seguir devem ser enviados com a solicitação:

### Autorização

Esse cabeçalho deverá ser enviado se você estiver usando a autenticação básica. Para obter mais informações, veja [Usando a autenticação básica HTTP com a REST API](#).

### ibm-mq-rest-csrf-token

Esse cabeçalho deve ser configurado, mas o valor pode ser qualquer coisa, incluindo ficar em branco.

Os cabeçalhos a seguir podem ser enviados opcionalmente com a solicitação:

### Aceitar-Conjunto de caracteres

Esse cabeçalho pode ser usado para indicar qual conjunto de caracteres é aceitável para a resposta. Se for especificado, esse cabeçalho deverá ser configurado como UTF-8.

### Accept-Language

Este cabeçalho especifica o idioma necessário para quaisquer exceções ou mensagens de erro retornados no corpo da mensagem de resposta.

### Formato do corpo da solicitação:


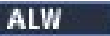

Nenhum.

### Requisitos de segurança


O responsável pela chamada deve ser autenticado no servidor mqweb. As funções MQWebAdmin e MQWebAdminRO não são aplicáveis à messaging REST API. Para obter mais informações sobre segurança para o REST API, consulte [Segurança do IBM MQ Console e do REST API](#).

Depois de autenticado para o servidor mqWeb, o usuário é capaz de usar o messaging REST API e o administrative REST API.

O principal de segurança do responsável pela chamada deve ter a capacidade de obter mensagens da fila especificada:

- A fila especificada pela parte *{queueName}* da URL do recurso precisa ser GET ativada.
-   Para a fila especificada pela parte *{queueName}* da URL do recurso, +GET, +INQ e a autoridade do +BROWSE autoridade precisa ser concedida à entidade de segurança do responsável pela chamada.
-  Para a fila especificada pela parte *{queueName}* da URL do recurso, UPDATE, o acesso precisa ser concedido à entidade de segurança do responsável pela chamada.

Esse proprietário de segurança pode ser o usuário que iniciou o servidor mqweb, ou o usuário que efetuou login no servidor mqweb. Se o gerenciador de filas ao qual você se conectar for um gerenciador de filas remotas, o principal de segurança poderá, em vez disso, ser o usuário fornecido pelas credenciais nas informações de conexão do gerenciador de filas remotas ou outro usuário determinado pelas regras de segurança do canal. Para obter mais informações, consulte [Determinando o principal de segurança usado pelo messaging REST API](#).

 No AIX, Linux, and Windows, é possível conceder autoridade às entidades de segurança para usar os recursos do IBM MQ usando o comando **setmqaut**. Para obter mais informações, consulte **setmqaut** (autoridade de concessão ou revogação).

 No z/OS, consulte [Configurando a segurança no z/OS](#).

### Códigos de status de resposta.

#### 200

Mensagem recebida com êxito.

#### 204

Não há mensagens disponíveis.

#### 400

Dados inválidos fornecidos.

Por exemplo, foi especificado um valor de parâmetro de consulta inválido.

#### 401

Não autenticado.

O ouvinte precisa ser autenticado para o servidor mqweb e ser um membro de um ou mais das funções MQWebAdmin, MQWebAdminRO ou MQWebUser. O cabeçalho `ibm-mq-rest-csrf-token`

também deve ser especificado.. Para obter mais informações, consulte [“Requisitos de segurança” na página 2255](#).

#### 403

Não autorizado.

O responsável pela chamadas é autenticado para o servidor mqweb e está associado a um proprietário válido No entanto, o proprietário não tem acesso a todos ou a um subconjunto dos recursos necessários do IBM MQ ou não está na função MQWebUser .. Para obter mais informações sobre o acesso necessário, consulte [“Requisitos de segurança” na página 2255](#).

#### 404

A fila não existe.

#### 405

A fila é inibida por GET

#### 500

Problema do servidor ou código de erro de IBM MQ.

#### 501

A resposta de HTTP não pôde ser construída

Por exemplo, a mensagem recebida tem um tipo incorreto ou tem o tipo correto, mas o corpo não pôde ser processado

#### 502

O principal de segurança atual não pode receber a mensagem porque o provedor de sistemas de mensagens não suporta a função necessária. Por exemplo, se o caminho da classe do servidor mqweb for inválido,

#### 503

Gerenciador de filas não em execução.

## Cabeçalhos de resposta

Os seguintes cabeçalhos são retornados com a resposta:

### Conteúdo-Idioma

Especifica o identificador de idioma da mensagem de resposta no caso de quaisquer erros ou exceções. Usado em conjunto com o cabeçalho da solicitação Accept - Language para indicar o idioma necessário para quaisquer condições de erro ou de exceção O padrão do servidor mqweb será usado se o idioma solicitado não for suportado

### Content-Length

Especifica o comprimento do corpo de resposta HTTP , mesmo quando não há conteúdo.. O valor contém o comprimento (bytes) dos dados da mensagem.

### Content-Type

Especifica o tipo de conteúdo retornado no corpo de resposta da mensagem recebida Após o sucesso, o valor é `text/plain; charset=utf-8`. No caso de erros ou exceções, o valor será `application/json; charset=utf-8`.

### **ibm-mq-md-correlationId**

Especifica o ID de correlação da mensagem recebida. O cabeçalho será retornado se a mensagem recebida contiver um ID de correlação válido. É representado como uma sequência codificada hexadecimal de 48 caracteres, representando 24 bytes.

Por exemplo:

```
ibm-mq-md-correlationId: 414d5120514d4144455620202020202067d8bf5923582e02
```

### **ibm-mq-md-correlationId**

Especifica o ID de correlação da mensagem recebida. O cabeçalho será retornado se a mensagem recebida contiver um ID de correlação válido. O ID de correlação pode ter um dos seguintes formatos:

- Uma sequência codificada hexadecimal de 48 caracteres, representando 24 bytes, prefixada com a sequência "ID: ". Por exemplo:

```
ibm-mq-md-correlationId: ID:414d5120514d4144455620202020202067d8bf5923582e02
```

- Um valor específico do aplicativo. O valor é uma sequência específica do aplicativo:

```
ibm-mq-md-correlationId: My-Custom-CorrelId
```

### **ibm-mq-md-expiração**

Especifica a duração de expiração restante de uma mensagem recebida O cabeçalho pode ser um dos seguintes valores:

#### **ilimitado**

A mensagem não expira.

#### **Valor de número inteiro**

Os milissegundos restantes antes da expiração da mensagem

### **REST API V1** **REST API V2** **ibm-mq-md-messageId**

Especifica o ID de mensagem que é alocado por IBM MQ para esta mensagem.. Como o cabeçalho `ibm-mq-md-correlationId`, ele é representado como uma sequência codificada hexadecimal de 48 caracteres, representando 24 bytes..

Por exemplo:

```
ibm-mq-md-messageId: 414d5120514d4144455620202020202067d8ce5923582f07
```

### **V 9.3.0** **REST API V3** **ibm-mq-md-messageId**

Especifica o ID de mensagem que é alocado por IBM MQ para esta mensagem.. Como o cabeçalho `ibm-mq-md-correlationId`, ele é representado como uma sequência codificada hexadecimal de 48 caracteres, representando 24 bytes, prefixados com a sequência "ID: "

Por exemplo:

```
ibm-mq-md-messageId: ID:414d5120514d4144455620202020202067d8ce5923582f07
```

### **ibm-mq-md-persistence**

Especifica a persistência da mensagem recebida.. O cabeçalho pode ser um dos seguintes valores:

#### **nonPersistent**

A mensagem não sobrevive a falhas do sistema ou reinícios do gerenciador de filas.

#### **persistente**

A mensagem sobrevive a falhas do sistema ou reinícios do gerenciador de filas.

### **V 9.3.0** **REST API V3** **ibm-mq-md-priority**

Retorna a configuração da prioridade da mensagem Por exemplo:

```
ibm-mq-md-priority: 3
```

### **ibm-mq-md-replyTo**

Especifica o destino de resposta para a mensagem recebida O formato do cabeçalho usa a notação padrão da fila de resposta e do gerenciador de fila, `replyQueue@replyQMgr`

Por exemplo:

```
ibm-mq-md-replyTo: myReplyQueue@myReplyQMgr
```

### V 9.3.3 **ibm-mq-resolved-qmgr**

Especifica o nome do gerenciador de filas que foi usado para a solicitação. Se um nome exclusivo foi usado na URL do recurso, esse cabeçalho identifica o nome do gerenciador de filas que está associado a esse nome exclusivo. Se o nome exclusivo usado no recurso URL se referir a um grupo de gerenciadores de filas, esse cabeçalho identificará qual gerenciador de filas dentro do grupo foi usado.

### V 9.3.0 **REST API V3** **ibm-mq-usr**

Retorna as propriedades definidas pelo usuário da mensagem. Diversas propriedades podem ser configuradas em uma mensagem, nesse caso, haverá duas ou mais instâncias separadas do cabeçalho de resposta `ibm-mq-usr`.

Por exemplo:

```
ibm-mq-usr: myIPprop;5;short
ibm-mq-usr: mySProp;"hi";string
ibm-mq-usr: myBProp>true;boolean
```

As propriedades têm a seguinte sintaxe:

```
ibm-mq-usr: property_name; user_value; user_type
```

#### **property\_name**

O nome da propriedade do usuário que está sendo especificada. Este deve ser um nome de propriedade JMS válido.

#### **user\_value**

O valor da propriedade.

#### **user\_type**

O tipo da propriedade:

- `boolean` (true/false, MQBOOL)
- `byte` (número inteiro de 8 bits, MQINT8)
- `short` (número inteiro de 16 bits, MQINT16)
- `integer` (número inteiro de 32 bits, MQINT32)
- `long` (número inteiro de 64 bits, MQINT64)
- `float` (32 bits reais, MQFLOAT32)
- `double` (real de 64 bits, MQFLOAT64)
- `string` (sequência entre aspas)

## **Formato do corpo de resposta**

Após o sucesso, o corpo de resposta contém o corpo da mensagem da mensagem recebida. Se ocorrer um erro, o corpo de resposta conterá uma mensagem de erro formatada por JSON. Ambas as respostas são codificadas por UTF-8. Para obter mais informações, consulte [REST API manipulação de erros](#).

Esteja ciente de que ao receber uma mensagem apenas IBM MQ MQSTR e JMS `TextMessage` mensagens formatadas são suportadas. Subsequentemente, todas as mensagens são recebidas sob o ponto de sincronização e quaisquer mensagens não manipuladas são deixadas na fila. A fila do IBM MQ pode ser configurada para mover essas mensagens suspeitas para um destino alternativo. Para obter informações adicionais, veja [Manipulando mensagens suspeitas em classes do IBM MQ para JMS](#).

## **Exemplos**

Os exemplos a seguir usam a URL de recurso v2. Se você estiver usando uma versão do IBM MQ anterior ao IBM MQ 9.1.5, use a URL do recurso v1 como alternativa. Ou seja, na URL do recurso, substitua v1 em que a URL de exemplo usa v2.

O exemplo a seguir registra um usuário chamado mquser com a senha mquser. Em cURL, a solicitação de login pode ser semelhante à seguinte Windows exemplo. O token LTPA é armazenado no arquivo cookiejar.txt usando a sinalização -c :

```
curl -k "https://localhost:9443/ibmmq/rest/v2/login" -X POST
-H "Content-Type: application/json" --data "{\"username\":\"mquser\",\"password\":\"mquser\"}"
-c c:\cookiejar.txt
```

Após o usuário efetuar login, o token LTPA e o cabeçalho ibm-mq-rest-csrf-token HTTP são usados para autenticar solicitações adicionais. O ibm-mq-rest-csrf-token token\_value pode ser qualquer valor, incluindo em branco.

- O exemplo de Windows cURL a seguir remove a próxima mensagem disponível da fila Q1 no gerenciador de filas QM1, usando opções padrão:

```
curl -k "https://localhost:9443/ibmmq/rest/v2/messaging/qmgr/QM1/queue/Q1/message"
-X DELETE -b c:\cookiejar.txt -H "ibm-mq-rest-csrf-token: token-value"
-H "Accept: text/plain"
```

- O exemplo de Windows cURL a seguir remove uma mensagem com um ID de correlação específico, 00abcdabcd, da fila Q1 no gerenciador de filas QM1:

```
curl -k "https://localhost:9443/ibmmq/rest/v2/messaging/qmgr/QM1/queue/Q1/message?
correlationId=0000000000000000000000000000000000000000000000000000000000000000abcdabcd"
-X DELETE -b c:\cookiejar.txt -H "ibm-mq-rest-csrf-token: token-value"
-H "Accept: text/plain"
```

- O exemplo de Windows cURL a seguir remove uma mensagem com um ID de correlação específico, 00abcdabcd, da fila Q1 no gerenciador de filas QM1, esperando até 30 segundos para que a mensagem se tornasse disponível.. Se 30 segundos passar sem que a mensagem especificada seja colocada na fila, a chamada DELETE retornará sem uma mensagem:

```
curl -k "https://localhost:9443/ibmmq/rest/v2/messaging/qmgr/QM1/queue/Q1/message?
correlationId=0000000000000000000000000000000000000000000000000000000000000000abcdabcd&wait=30000"
-X DELETE -b c:\cookiejar.txt -H "ibm-mq-rest-csrf-token: token-value"
-H "Accept: text/plain"
```

## **/messaging/qmgr/{qmgrName}/queue/{queueName}/messagelist**

É possível usar o método GET HTTP com o recurso /messaging/qmgr/{qmgrName}/queue/{queueName}/messagelist para obter uma lista de mensagens disponíveis da fila especificada no gerenciador de filas especificado.

### ***GET /messaging/qmgr/{qmgrName}/queue/{queueName}/messagelist***

É possível usar o método GET HTTP com o recurso /messaging/qmgr/{qmgrName}/queue/{queueName}/messagelist para obter uma lista de mensagens disponíveis da fila especificada no gerenciador de filas especificado.

Procura uma lista de resumo de mensagens do gerenciador de filas e da fila especificados. O gerenciador de filas deve estar na mesma máquina que o servidor mqweb. Os dados de resumo são retornados no corpo de resposta HTTP como uma matriz formatada por JSON Os dados não contêm a carga útil das mensagens e são recebidos usando o contexto do usuário atual Nenhuma mensagem é removida da fila associada

Se uma solicitação for feita para obter uma lista de mensagens disponíveis de uma fila que é inibida por GET, uma matriz JSON vazia será retornada.

- [“URL do recurso” na página 2260](#)
- [“Parâmetros de consulta opcionais” na página 2260](#)
- [“Cabeçalhos de solicitação” na página 2262](#)
- [“Formato do corpo da solicitação:” na página 2262](#)
- [“Requisitos de segurança” na página 2262](#)

- “Códigos de status de resposta.” na página 2262
- “Cabeçalhos de resposta” na página 2263
- “Formato do corpo de resposta” na página 2263
- “Exemplos” na página 2264

## URL do recurso

`https://host:port/ibmmq/rest/v1/messaging/qmgr/{qmgrName}/queue/{queueName}/messagelist`

`https://host:port/ibmmq/rest/v2/messaging/qmgr/{qmgrName}/queue/{queueName}/messagelist`

**V 9.3.0** `https://host:port/ibmmq/rest/v3/messaging/qmgr/{qmgrName}/queue/{queueName}/messagelist`

### qmgrName

Especifica o nome do gerenciador de filas ao qual se conectar para sistema de mensagens. O gerenciador de filas deve estar na mesma máquina que o servidor mqweb.

**V 9.3.3** A partir do IBM MQ 9.3.3, é possível conectar-se a um gerenciador de filas locais ou a um gerenciador de filas remotas. O nome especificado para **qmgrName** depende de como seu servidor mqweb está configurado:

- Se o seu servidor mqweb estiver configurado para se conectar apenas aos gerenciadores de filas locais, use o nome do gerenciador de filas.
- Se o seu servidor mqweb estiver configurado para conectar gerenciadores de filas remotas, use o nome exclusivo para o gerenciador de filas, conforme especificado nas informações de conexão do gerenciador de filas remotas

É possível determinar se seu servidor mqweb está configurado para se conectar a gerenciadores de fila locais ou gerenciadores de filas remotas usando o comando **dspmweb properties** e visualizando a propriedade **mqRestMessagingConnectionMode**.

O nome do gerenciador de filas faz distinção entre maiúsculas e minúsculas.

Se o nome do gerenciador de filas incluir uma barra, um ponto ou um sinal de percentual, esses caracteres deverão ser codificados para URL:

- Uma barra (/) precisa ser codificada como %2F.
- Um sinal de percentual (%) precisa ser codificado como %25.

### queueName

Especifica o nome da fila a partir da qual procurar as mensagens.

A fila deve estar definida como sendo local ou um alias que aponte para uma fila local.

O nome da fila faz distinção entre maiúsculas e minúsculas..

Se o nome da fila incluir uma barra ou um sinal de porcentagem, esses caracteres deverão ser codificados em URL :

- Uma barra, /, deve ser codificada como %2F.
- Um sinal de porcentagem, %, deve ser codificado como %25.

Será possível usar HTTP (Protocolo de Transporte de Hipertexto) no lugar de HTTPS se você ativar as conexões HTTP. Para obter mais informações sobre como ativar HTTP, veja [Configurando as portas HTTP e HTTPS](#).

## Parâmetros de consulta opcionais

**REST API V2** `correlationId=hexValue`

Especifica que o método HTTP retorna a próxima mensagem com o ID de correlação correspondente.



### hexValue

O parâmetro de consulta deve ser especificado como uma cadeia codificada hexadecimal de 48 caracteres, representando 24 bytes..

Por exemplo:

```
../messagelist?correlationId=414d5120514d41444556202020202067d8bf5923582e02
```

V 9.3.0

REST API V3

**correlationId= ID:hexValue** ou **correlationId=application\_specific\_value**

Especifica que o método HTTP retorna uma lista de mensagens com o ID de correlação correspondente.

### hexValue

O parâmetro de consulta deve ser especificado como uma sequência codificada hexadecimal de 48 caracteres, representando 24 bytes, e precedido pela sequência "ID: "

Por exemplo:

```
../message?correlationId=ID:414d5120514d41444556202020202067d8bf5923582e02
```

### application\_specific\_value

O parâmetro de consulta pode ser especificado como uma sequência específica do aplicativo

Por exemplo:

```
../message?correlationId=My-Custom-CorrelId
```

REST API V1

REST API V2

**messageId=hexValue**

Especifica que o método HTTP retorna a próxima mensagem com o ID de mensagem correspondente.

### hexValue

O parâmetro de consulta deve ser especificado como uma cadeia codificada hexadecimal de 48 caracteres, representando 24 bytes..

Por exemplo:

```
../message?messageId=414d5120514d41444556202020202067d8ce5923582f07
```

V 9.3.0

REST API V3

**messageId= ID:hexValue**

Especifica que o método HTTP retorna a próxima mensagem com o ID de mensagem correspondente.

### hexValue

O parâmetro de consulta deve ser especificado como uma sequência codificada hexadecimal de 48 caracteres, representando 24 bytes, e precedido pela sequência "ID: "

Por exemplo:

```
../message?messageId=ID:414d5120514d41444556202020202067d8ce5923582f07
```

### limit=integerValue

Especifica que o corpo de resposta do método HTTP é limitado a elementos JSON *integerValue*

### integerValue

O parâmetro de consulta deve ser especificado como um valor inteiro que representa o número máximo de elementos que estão contidos no corpo de resposta JSON.

O valor padrão é 10 e o valor máximo é 2147483647.

Por exemplo:

```
../messagelist?limit=250
```

## Cabeçalhos de solicitação

Os cabeçalhos a seguir devem ser enviados com a solicitação:

### Autorização

Esse cabeçalho deverá ser enviado se você estiver usando a autenticação básica. Para obter mais informações, veja [Usando a autenticação básica HTTP com a REST API](#).

### ibm-mq-rest-csrf-token

Esse cabeçalho deve ser configurado, mas o valor pode ser qualquer coisa, incluindo ficar em branco.

Os cabeçalhos a seguir podem ser enviados opcionalmente com a solicitação:

### Aceitar-Conjunto de caracteres

Esse cabeçalho pode ser usado para indicar qual conjunto de caracteres é aceitável para a resposta. Se for especificado, esse cabeçalho deverá ser configurado como UTF-8.

### Accept-Language

Este cabeçalho especifica o idioma necessário para quaisquer exceções ou mensagens de erro retornados no corpo da mensagem de resposta.

## Formato do corpo da solicitação:




Nenhum.

## Requisitos de segurança


O responsável pela chamada deve ser autenticado no servidor mqweb. As funções MQWebAdmin e MQWebAdminRO não são aplicáveis à messaging REST API. Para obter mais informações sobre segurança para o REST API, consulte [Segurança do IBM MQ Console e do REST API](#).

Depois de autenticado para o servidor mqWeb, o usuário é capaz de usar o messaging REST API e o administrative REST API.

A capacidade de procurar mensagens da fila especificada deve ser concedida à entidade de segurança do responsável pela chamada:

- A fila especificada pela parte *{queueName}* da URL do recurso precisa ser BROWSE ativada.
-   Para a fila especificada pela parte *{queueName}* da URL do recurso, +GET, +INQ e a autoridade do +BROWSE autoridade precisa ser concedida à entidade de segurança do responsável pela chamada.
-  Para a fila especificada pela parte *{queueName}* da URL do recurso, UPDATE, o acesso precisa ser concedido à entidade de segurança do responsável pela chamada.

Esse proprietário de segurança pode ser o usuário que iniciou o servidor mqweb, ou o usuário que efetuou login no servidor mqweb. Se o gerenciador de filas ao qual você se conectar for um gerenciador de filas remotas, o principal de segurança poderá, em vez disso, ser o usuário fornecido pelas credenciais nas informações de conexão do gerenciador de filas remotas ou outro usuário determinado pelas regras de segurança do canal. Para obter mais informações, consulte [Determinando o principal de segurança usado pelo messaging REST API](#).

 No AIX, Linux, and Windows, é possível conceder autoridade às entidades de segurança para usar os recursos do IBM MQ usando o comando **setmqaut**. Para obter mais informações, consulte [setmqaut](#) (autoridade de concessão ou revogação).

 No z/OS, consulte [Configurando a segurança no z/OS](#).

## Códigos de status de resposta.

### 200

Lista de mensagem recebida com êxito

**400**

Dados inválidos fornecidos.

Por exemplo, foi especificado um valor de parâmetro de consulta inválido.

**401**

Não autenticado.

O ouvinte precisa ser autenticado para o servidor mqweb e ser um membro de um ou mais das funções MQWebAdmin, MQWebAdminRO ou MQWebUser. O cabeçalho `ibm-mq-rest-csrf-token` também deve ser especificado.. Para obter mais informações, consulte [“Requisitos de segurança” na página 2262](#).

**403**

Não autorizado.

O responsável pela chamadas é autenticado para o servidor mqweb e está associado a um proprietário válido No entanto, o proprietário não tem acesso a todos ou a um subconjunto dos recursos necessários do IBM MQ ou não está na função MQWebUser .. Para obter mais informações sobre o acesso necessário, consulte [“Requisitos de segurança” na página 2262](#).

**404**

A fila não existe.

**500**

Problema do servidor ou código de erro de IBM MQ.

**501**

A resposta de HTTP não pôde ser construída

Por exemplo, a mensagem recebida tem um tipo incorreto ou tem o tipo correto, mas o corpo não pôde ser processado

**502**

O principal de segurança atual não pode receber a mensagem porque o provedor de sistemas de mensagens não suporta a função necessária. Por exemplo, se o caminho da classe do servidor mqweb for inválido,

**503**

Gerenciador de filas não em execução.

## Cabeçalhos de resposta

### Conteúdo-Idioma

Especifica o identificador de idioma da mensagem de resposta no caso de quaisquer erros ou exceções. Usado em conjunto com o cabeçalho da solicitação `Accept-Language` para indicar o idioma necessário para quaisquer condições de erro ou de exceção O padrão do servidor mqweb será usado se o idioma solicitado não for suportado

### Content-Length

Especifica o comprimento do corpo de resposta HTTP , mesmo quando não há conteúdo.. O valor contém o comprimento dos dados da mensagem, em bytes..

### Content-Type

Especifica o tipo de corpo de resposta O valor é `application/json; charset=utf-8`.

### **V 9.3.3** `ibm-mq-resolved-qmgr`

Especifica o nome do gerenciador de filas que foi usado para a solicitação Se um nome exclusivo foi usado na URLdo recurso, esse cabeçalho identifica o nome do gerenciador de filas que está associado a esse nome exclusivo. Se o nome exclusivo usado no recurso URL se referir a um grupo de gerenciadores de filas, esse cabeçalho identificará qual gerenciador de filas dentro do grupo foi usado.

## Formato do corpo de resposta

Após o sucesso, o corpo de resposta é uma resposta codificada por UTF-8 A resposta contém um objeto JSON externo que contém uma única matriz JSON chamada `messages`.. Cada elemento na matriz é um

objeto JSON que contém informações sobre uma mensagem na fila.. Cada elemento contém os seguintes atributos:

**REST API V1** **REST API V2** **correlationId**

Especifica o ID de correlação da mensagem.. O valor será retornado se a mensagem contiver um ID de correlação válido.

**V 9.3.0** **REST API V3** **correlationId**

Especifica o ID de correlação da mensagem.. O valor será retornado se a mensagem contiver um ID de correlação válido. O ID de correlação é prefixado com a sequência "ID:" ou pode ser um valor específico de aplicativo.

**REST API V1** **REST API V2** **messageId**

Especifica o ID de mensagem alocado pelo IBM MQ para essa mensagem. É representado como uma sequência codificada hexadecimal de 48 caracteres, representando 24 bytes.

**V 9.3.0** **REST API V3** **messageId**

Especifica o ID de mensagem alocado pelo IBM MQ para essa mensagem. É representado como uma sequência codificada hexadecimal de 48 caracteres, representando 24 bytes. O ID de mensagem é prefixado com a sequência "ID:".

### format

Especifica o campo de formato do MQMD Em circunstâncias normais, mensagens de texto conterão o valor IBM MQ MQSTR .

Se for feita uma solicitação para obter uma lista de mensagens em uma fila inibida por GET, uma matriz JSON vazia será retornada.

Se ocorrer um erro, o corpo de resposta conterá uma mensagem de erro formatada por JSON Para obter mais informações, consulte [REST API manipulação de erros](#).

## Exemplos

Os exemplos a seguir usam a URL de recurso v2. Se você estiver usando uma versão do IBM MQ anterior ao IBM MQ 9.1.5, use a URL do recurso v1 como alternativa. Ou seja, na URL do recurso, substitua v1 em que a URL de exemplo usa v2.

O exemplo a seguir registra um usuário chamado mquser com a senha mquser. Em cURL, a solicitação de login pode ser semelhante à seguinte Windows exemplo. O token LTPA é armazenado no arquivo cookiejar.txt usando a sinalização -c :

```
curl -k "https://localhost:9443/ibmmq/rest/v2/login" -X POST
-H "Content-Type: application/json" --data "{\"username\":\"mquser\", \"password\":\"mquser\"}"
-c c:\cookiejar.txt
```

Após o usuário efetuar login, o token LTPA e o cabeçalho `ibm-mq-rest-csrf-token` HTTP são usados para autenticar solicitações adicionais. O `ibm-mq-rest-csrf-token` `token_value` pode ser qualquer valor, incluindo em branco.

- O exemplo de Windows cURL a seguir lista as próximas dez mensagens disponíveis da fila Q1 no gerenciador de filas QM1, usando as opções padrão:

```
curl -k "https://localhost:9443/ibmmq/rest/v2/messaging/qmgr/QM1/queue/Q1/messagelist"
-X GET -b c:\cookiejar.txt -H "ibm-mq-rest-csrf-token: token-value"
-H "Accept: application/json"
```

- O exemplo de Windows cURL a seguir lista as próximas 200 mensagens disponíveis da fila Q1 no gerenciador de filas QM1, usando opções padrão:

```
curl -k "https://localhost:9443/ibmmq/rest/v2/messaging/qmgr/QM1/queue/Q1/messagelist?
limit=200"
-X GET -b c:\cookiejar.txt -H "ibm-mq-rest-csrf-token: token-value"
-H "Accept: application/json"
```



**V 9.3.0** `https://host:port/ibmmq/rest/v3/messaging/qmgr/{qmgrName}/topic/{topicString}/message`

### **qmgrName**

Especifica o nome de um gerenciador de filas ao qual se conectar para o sistema de mensagens

**V 9.3.3** A partir do IBM MQ 9.3.3, é possível conectar-se a um gerenciador de filas locais ou a um gerenciador de filas remotas O nome especificado para **qmgrName** depende de como seu servidor mqweb está configurado:

- Se o seu servidor mqweb estiver configurado para se conectar apenas aos gerenciadores de filas locais, use o nome do gerenciador de filas.
- Se o seu servidor mqweb estiver configurado para conectar gerenciadores de filas remotas, use o nome exclusivo para o gerenciador de filas, conforme especificado nas informações de conexão do gerenciador de filas remotas

É possível determinar se seu servidor mqweb está configurado para se conectar a gerenciadores de fila locais ou gerenciadores de filas remotas usando o comando **dspmweb properties** e visualizando a propriedade **mqRestMessagingConnectionMode** .

O nome faz distinção entre maiúsculas e minúsculas.

Se o nome incluir uma barra, um ponto ou um sinal de porcentagem, esses caracteres deverão ser codificados em URL :

- Uma barra deve ser codificada como %2F.
- Um período deve ser codificado como %2E.
- Um sinal de percentual deve ser codificado como %25.

### **topicString**

Especifica a sequência de tópicos na qual publicar a mensagem..

A sequência de tópicos faz distinção entre maiúsculas e minúsculas. A sequência de tópicos pode conter vários níveis de tópicos separados pelo delimitador de barra.

Se a sequência de tópicos contiver um sinal de porcentagem, um ponto final ou um ponto de interrogação, esses caracteres deverão ser codificados em URL :

- Um sinal de percentual deve ser codificado como %25.
- Um período deve ser codificado como %2E.
- Um ponto de interrogação deve ser codificado como %3F.

Se a sequência de tópicos iniciar ou terminar com uma barra, ela deverá ser codificada com um %2F.

Por exemplo, para publicar na sequência de tópicos:

- `sport/football` no gerenciador de filas MY.QMGR, você usa a URLa seguir:

```
https://localhost:9443/ibmmq/rest/v2/messaging/qmgr/MY%2EQMGR/topic/sport/football/message
```

- `/sport/football` no gerenciador de filas MY.QMGR, você usa a URLa seguir:

```
https://localhost:9443/ibmmq/rest/v2/messaging/qmgr/MY%2EQMGR/topic/%2Fsport/football/message
```

Será possível usar HTTP (Protocolo de Transporte de Hipertexto) no lugar de HTTPS se você ativar as conexões HTTP. Para obter mais informações sobre como ativar HTTP, veja [Configurando as portas HTTP e HTTPS](#).

## **Cabeçalhos de solicitação**

Os cabeçalhos a seguir devem ser enviados com a solicitação:

## Autorização

Esse cabeçalho deverá ser enviado se você estiver usando a autenticação básica. Para obter mais informações, veja [Usando a autenticação básica HTTP com a REST API](#).

## Content-Type

Esse cabeçalho deve ser enviado com um dos seguintes valores:

- `text/plain;charset=utf-8`
- `text/html;charset=utf-8`
- `text/xml;charset=utf-8`
- `application/json;charset=utf-8`
- `application/xml;charset=utf-8`

## ibm-mq-rest-csrf-token

Esse cabeçalho deve ser configurado, mas o valor pode ser qualquer coisa, incluindo ficar em branco.

Os cabeçalhos a seguir podem ser enviados opcionalmente com a solicitação:

## Accept-Language

Este cabeçalho especifica o idioma necessário para quaisquer exceções ou mensagens de erro retornados no corpo da mensagem de resposta

## ibm-mq-md-expiração

Este cabeçalho configura a duração de expiração para a mensagem criada. A expiração de uma mensagem é iniciada a partir do momento em que a mensagem chega ao gerenciador de filas. Como resultado, a latência de rede é ignorada. O cabeçalho deve ser especificado como um dos seguintes valores:

### ilimitado

A mensagem não expira.

Esse valor é o valor padrão.

### Valor de número inteiro

Milissegundos antes da expiração da mensagem

Limitado ao intervalo de 0 a 99999999900.

## ibm-mq-md-persistence

Esse cabeçalho configura a persistência para a mensagem criada. O cabeçalho deve ser especificado como um dos seguintes valores:

### nonPersistent

A mensagem não sobrevive a falhas do sistema ou reinícios do gerenciador de filas.

Esse valor é o valor padrão.

### persistente

A mensagem sobrevive a falhas do sistema ou reinícios do gerenciador de filas.

V 9.3.0

REST API V3

## ibm-mq-md-priority

Esse cabeçalho configura a prioridade da mensagem criada. O cabeçalho deve ser especificado como um dos seguintes valores:

### asDestination

A mensagem usa a prioridade especificada no atributo DEFPRTY do objeto de fila IBM MQ subjacente.

### Valor de número inteiro

Especifique a prioridade real como um número inteiro no intervalo de 0 a 9.

Por exemplo:

```
ibm-mq-md-priority: asDestination
```

## ibm-mq-md-replyTo

Este cabeçalho configura o destino de resposta para a mensagem criada. O formato do cabeçalho usa a notação padrão de fornecer a fila de respostas e um gerenciador de filas opcional:

```
replyQueue[@replyQmgr]
```

Por exemplo:

```
ibm-mq-md-replyTo: myReplyQueue@myReplyQMGR
```

## V 9.3.0 REST API V3 **ibm-mq-usr**

Configure as propriedades definidas pelo usuário da mensagem de solicitação. Várias propriedades podem ser configuradas em uma mensagem. É possível especificar várias propriedades separadas por vírgula em um único cabeçalho de solicitação `ibm-mq-usr` ou é possível usar duas ou mais instâncias separadas do cabeçalho de solicitação `ibm-mq-usr`.

Por exemplo:

```
ibm-mq-usr: myIPprop;5;short
ibm-mq-usr: mySPprop;"hi";string
ibm-mq-usr: myBProp>true;boolean
ibm-mq-usr: myA;5;byte,myB;-10;integer
```

As propriedades têm a seguinte sintaxe:

```
ibm-mq-usr: property_name; user_value; user_type
```

### **property\_name**

O nome da propriedade do usuário que está sendo especificada. Este deve ser um nome de propriedade JMS válido.

### **user\_value**

O valor da propriedade.

### **user\_type**

O tipo da propriedade:

- `boolean` (true/false, MQBOOL)
- `byte` (número inteiro de 8 bits, MQINT8)
- `short` (número inteiro de 16 bits, MQINT16)
- `integer` (número inteiro de 32 bits, MQINT32)
- `long` (número inteiro de 64 bits, MQINT64)
- `float` (32 bits reais, MQFLOAT32)
- `double` (real de 64 bits, MQFLOAT64)
- `string` (sequência entre aspas)

## Formato do corpo da solicitação:

O corpo da solicitação deve ser de texto e usar a codificação UTF-8. Não é necessária uma estrutura de texto específica. Uma mensagem formatada MQSTR que contém o texto do corpo da solicitação é criada e publicada no tópico especificado.

V 9.3.0 REST API V3 Se as propriedades definidas pelo usuário da API REST V3 ou os recursos de ID de correlação específicos do aplicativo forem usados, uma mensagem formatada JMS `TextMessage` contendo o texto do corpo da solicitação será criada e colocada na fila especificada.

Para obter mais informações, consulte [exemplos](#).






## Requisitos de segurança


O responsável pela chamada deve ser autenticado no servidor mqweb. As funções MQWebAdmin e MQWebAdminRO não são aplicáveis à messaging REST API. Para obter mais informações sobre segurança para o REST API, consulte [Segurança do IBM MQ Console e do REST API](#).

Depois de autenticado para o servidor mqWeb, o usuário é capaz de usar o messaging REST API e o administrative REST API.

O proprietário de segurança do responsável pela chamada deve ter a capacidade de publicar mensagens no tópico especificado:

- O tópico especificado pela parte *{topicString}* do recurso URL deve ser PUBLISH ativado.
-   Para o tópico especificado pela parte *{topicString}* do recurso URL, a autoridade +PUB deve ser concedida ao principal de segurança do responsável pela chamada.
-  Para o tópico especificado pela parte *{topicString}* do recurso URL, UPDATE o acesso deve ser concedido ao principal de segurança do responsável pela chamada.

Esse proprietário de segurança pode ser o usuário que iniciou o servidor mqweb, ou o usuário que efetuou login no servidor mqweb. Se o gerenciador de filas ao qual você se conectar for um gerenciador de filas remotas, o principal de segurança poderá, em vez disso, ser o usuário fornecido pelas credenciais nas informações de conexão do gerenciador de filas remotas ou outro usuário determinado pelas regras de segurança do canal. Para obter mais informações, consulte [Determinando o principal de segurança usado pelo messaging REST API](#).

 No AIX, Linux, and Windows, é possível conceder autoridade às entidades de segurança para usar os recursos do IBM MQ usando o comando **setmqaut**. Para obter mais informações, consulte [setmqaut](#) (autoridade de concessão ou revogação).

 No z/OS, consulte [Configurando a segurança no z/OS](#).

Se você usar Advanced Message Security (AMS) com o messaging REST API, observe que todas as mensagens serão criptografadas usando o contexto do servidor mqweb, não o contexto do usuário que postar a mensagem.

## Códigos de status de resposta.

### 201

Mensagem criada e publicada com sucesso.

### 400

Dados inválidos fornecidos.

Por exemplo, um valor de cabeçalho de solicitação inválido foi especificado.

### 401

Não autenticado.

O ouvinte precisa ser autenticado para o servidor mqweb e ser um membro de um ou mais das funções MQWebAdmin, MQWebAdminRO ou MQWebUser. O cabeçalho `ibm-mq-rest-csrf-token` também deve ser especificado.. Para obter mais informações, consulte [“Requisitos de segurança” na página 2269](#).

### 403

Não autorizado.

O responsável pela chamadas é autenticado para o servidor mqweb e está associado a um proprietário válido No entanto, o proprietário não tem acesso a todos ou a um subconjunto dos recursos necessários do IBM MQ ou não está na função MQWebUser .. Para obter mais informações sobre o acesso necessário, consulte [“Requisitos de segurança” na página 2269](#).

### 404

O gerenciador de filas não existe.

#### 405

O tópico é inibido por PUBLISH

#### 415

Um cabeçalho ou corpo da mensagem é um tipo de mídia não suportado..

Por exemplo, o cabeçalho Content-Type é configurado para um tipo de mídia não suportado

#### 500

Problema do servidor ou código de erro de IBM MQ.

#### 502

O proprietário de segurança atual não pode publicar a mensagem, pois o provedor de sistemas de mensagens não suporta a função necessária Por exemplo, se o caminho da classe do servidor mqweb for inválido,

#### 503

Gerenciador de filas não em execução.

## Cabeçalhos de resposta

Os seguintes cabeçalhos são retornados com a resposta:

### Conteúdo-Idioma

Especifica o identificador de idioma da mensagem de resposta no caso de quaisquer erros ou exceções. Usado em conjunto com o cabeçalho da solicitação Accept-Language para indicar o idioma necessário para quaisquer condições de erro ou de exceção O padrão do servidor mqweb será usado se o idioma solicitado não for suportado

### Content-Length

Especifica o comprimento do corpo de resposta HTTP , mesmo quando não há conteúdo.. Após o sucesso, o valor é zero.

### Content-Type

Especifica o tipo de corpo de resposta Após o sucesso, o valor é text/plain;charset=utf-8. No caso de erros ou exceções, o valor será application/json;charset=utf-8.

### V 9.3.3 ibm-mq-resolved-qmgr

Especifica o nome do gerenciador de filas que foi usado para a solicitação Se um nome exclusivo foi usado na URL do recurso, esse cabeçalho identifica o nome do gerenciador de filas que está associado a esse nome exclusivo. Se o nome exclusivo usado no recurso URL se referir a um grupo de gerenciadores de filas, esse cabeçalho identificará qual gerenciador de filas dentro do grupo foi usado.

## Formato do corpo de resposta

O corpo de resposta estará vazio se a mensagem for publicada com êxito.. Se ocorrer um erro, o corpo de resposta conterá uma mensagem de erro Para obter mais informações, consulte [REST API manipulação de erros](#).

## Exemplos

O exemplo a seguir registra um usuário chamado mquser com a senha mquser. Em cURL, a solicitação de login pode ser semelhante à seguinte Windows exemplo. O token LTPA é armazenado no arquivo cookiejar.txt usando a sinalização -c :

```
curl -k "https://localhost:9443/ibmmq/rest/v1/login" -X POST
-H "Content-Type: application/json" --data "{\"username\":\"mquser\", \"password\":\"mquser\"}"
-c c:\cookiejar.txt
```

Após o usuário efetuar login, o token LTPA e o cabeçalho ibm-mq-rest-csrf-token HTTP são usados para autenticar solicitações adicionais. O ibm-mq-rest-csrf-token token\_value pode ser qualquer valor, incluindo em branco.

- O exemplo Windows cURL a seguir publica uma mensagem na sequência de tópicos myTopic no gerenciador de filas QM1, usando opções padrão. A mensagem contém o texto *"Hello World!"*:

```
curl -k "https://localhost:9443/ibmmq/rest/v2/messaging/qmgr/QM1/topic/myTopic/message"
-X POST -b c:\cookiejar.txt -H "ibm-mq-rest-csrf-token: token_value"
-H "Content-Type: text/plain;charset=utf-8" --data "Hello World!"
```

- O exemplo Windows cURL a seguir publica uma mensagem persistente para a sequência de tópicos myTopic/thisTopic no gerenciador de filas QM1, com uma expiração de 2 minutos. A mensagem contém o texto *"Hello World!"*:

```
curl -k "https://localhost:9443/ibmmq/rest/v2/messaging/qmgr/QM1/topic/myTopic%2FthisTopic/
message"
-X POST -b c:\cookiejar.txt -H "ibm-mq-rest-csrf-token: token_value"
-H "Content-Type: text/plain;charset=utf-8" -H "ibm-mq-md-persistence: persistent"
-H "ibm-mq-md-expiry: 120000" --data "Hello World!"
```



## Avisos

---

Estas informações foram desenvolvidas para produtos e serviços oferecidos nos Estados Unidos.

É possível que a IBM não ofereça os produtos, serviços ou recursos discutidos nesta publicação em outros países. Consulte seu representante local do IBM para obter informações sobre produtos e serviços disponíveis atualmente em sua área. Qualquer referência a produtos, programas ou serviços IBM não significa que apenas produtos, programas ou serviços IBM possam ser utilizados. Qualquer outro produto, programa ou serviço, funcionalmente equivalente, poderá ser utilizado em substituição daqueles, desde que não infrinja nenhum direito de propriedade intelectual da IBM. Entretanto, a avaliação e verificação da operação de qualquer produto, programa ou serviço não IBM são de responsabilidade do Cliente.

A IBM pode ter patentes ou aplicativos de patentes pendentes relativas aos assuntos tratados nesta publicação. O fornecimento desta publicação não garante ao Cliente nenhum sobre tais patentes. É possível enviar pedidos de licença, por escrito, para:

Gerência de Relações Comerciais e Industriais da IBM Brasil  
Av. Pasteur, 138-146  
Botafogo  
Rio de Janeiro, RJ  
U.S.A.

Para pedidos de licença relacionados a informações de DBCS (Conjunto de Caracteres de Byte Duplo), entre em contato com o Departamento de Propriedade Intelectual da IBM em seu país ou envie pedidos de licença, por escrito, para:

Intellectual Property Licensing  
Legal and Intellectual Property Law  
IBM Japan, Ltd.  
19-21, Nihonbashi-Hakozakicho, Chuo-ku  
Tokyo 103-8510, Japan

**O parágrafo a seguir não se aplica a nenhum país em que tais disposições não estejam de acordo com a legislação local:** A INTERNATIONAL BUSINESS MACHINES CORPORATION FORNECE ESTA PUBLICAÇÃO "NO ESTADO EM QUE SE ENCONTRA", SEM GARANTIA DE NENHUM TIPO, SEJA EXPRESSA OU IMPLÍCITA, INCLUINDO, MAS A ELAS NÃO SE LIMITANDO, AS GARANTIAS IMPLÍCITAS DE NÃO INFRAÇÃO, COMERCIALIZAÇÃO OU ADEQUAÇÃO A UM DETERMINADO PROPÓSITO. Alguns países não permitem a exclusão de garantias expressas ou implícitas em certas transações; portanto, essa disposição pode não se aplicar ao Cliente.

Essas informações podem conter imprecisões técnicas ou erros tipográficos. São feitas alterações periódicas nas informações aqui contidas; tais alterações serão incorporadas em futuras edições desta publicação. IBM pode aperfeiçoar e/ou alterar no produto(s) e/ou programa(s) descritos nesta publicação a qualquer momento sem aviso prévio.

Referências nestas informações a websites não IBM são fornecidas apenas por conveniência e não representam de forma alguma um endosso a esses websites. Os materiais contidos nesses websites não fazem parte dos materiais desse produto IBM e a utilização desses websites é de inteira responsabilidade do Cliente.

A IBM pode utilizar ou distribuir as informações fornecidas da forma que julgar apropriada sem incorrer em qualquer obrigação para com o Cliente.

Os licenciados deste programa que desejarem obter informações sobre este assunto com o propósito de permitir: (i) a troca de informações entre programas criados independentemente e outros programas (incluindo este) e (ii) o uso mútuo das informações trocadas, deverão entrar em contato com:

Av. Pasteur, 138-146  
Av. Pasteur, 138-146

Botafogo  
Rio de Janeiro, RJ  
U.S.A.

Tais informações podem estar disponíveis, sujeitas a termos e condições apropriadas, incluindo em alguns casos o pagamento de uma taxa.

O programa licenciado descrito nesta publicação e todo o material licenciado disponível para ele são fornecidos pela IBM sob os termos do IBM Customer Agreement, IBM Contrato de Licença do Programa Internacional ou qualquer contrato equivalente entre as partes.

Todos os dados de desempenho aqui contidos foram determinados em um ambiente controlado. Portanto, os resultados obtidos em outros ambientes operacionais podem variar significativamente. Algumas medidas podem ter sido tomadas em sistemas em nível de desenvolvimento e não há garantia de que estas medidas serão iguais em sistemas geralmente disponíveis. Além disto, algumas medidas podem ter sido estimadas através de extrapolação. Os resultados reais podem variar. usuários deste documento devem verificar os dados aplicáveis para seu ambiente específico.

As informações relativas a produtos não IBM foram obtidas junto aos fornecedores dos respectivos produtos, de seus anúncios publicados ou de outras fontes disponíveis publicamente. A IBM não testou estes produtos e não pode confirmar a precisão de seu desempenho, compatibilidade nem qualquer outra reivindicação relacionada a produtos não IBM. Dúvidas sobre os recursos de produtos não IBM devem ser encaminhadas diretamente a seus fornecedores.

Todas as declarações relacionadas aos objetivos e intenções futuras da IBM estão sujeitas a alterações ou cancelamento sem aviso prévio e representam somente metas e objetivos.

Essas informações contêm exemplos de dados e relatórios utilizados em operações diárias de negócios. Para ilustrá-los da forma mais completa possível, os exemplos incluem nomes de indivíduos, empresas, marcas e produtos. Todos estes nomes são fictícios e qualquer semelhança com os nomes e endereços utilizados por uma empresa real é mera coincidência.

#### LICENÇA DE COPYRIGHT:

Estas informações contêm programas de aplicativos de amostra na linguagem fonte, ilustrando as técnicas de programação em diversas plataformas operacionais. O Cliente pode copiar, modificar e distribuir estes programas de amostra sem a necessidade de pagar à IBM, com objetivos de desenvolvimento, uso, marketing ou distribuição de programas aplicativos em conformidade com a interface de programação de aplicativo para a plataforma operacional para a qual os programas de amostra são criados. Esses exemplos não foram testados completamente em todas as condições. Portanto, a IBM não pode garantir ou implicar a confiabilidade, manutenção ou função destes programas.

Se estiver visualizando estas informações em formato eletrônico, as fotografias e ilustrações coloridas poderão não aparecer.

## Informações sobre a Interface de Programação

---

As informações da interface de programação, se fornecidas, destinam-se a ajudá-lo a criar software aplicativo para uso com este programa.

Este manual contém informações sobre interfaces de programação desejadas que permitem que o cliente grave programas para obter os serviços do WebSphere MQ.

No entanto, estas informações também podem conter informações sobre diagnósticos, modificações e ajustes. As informações sobre diagnósticos, modificações e ajustes são fornecidas para ajudá-lo a depurar seu software aplicativo.

**Importante:** Não use essas informações de diagnóstico, modificação e ajuste como uma interface de programação, pois elas estão sujeitas a mudanças

## Marcas comerciais

---

IBM, o logotipo IBM , ibm.com, são marcas registradas da IBM Corporation, registradas em várias jurisdições no mundo todo Uma lista atual de marcas registradas da IBM está disponível na Web em "Informações de copyright e marca registrada" [www.ibm.com/legal/copytrade.shtml](http://www.ibm.com/legal/copytrade.shtml). Outros nomes de produtos e serviços podem ser marcas comerciais da IBM ou de outras empresas.

Microsoft e Windows são marcas registradas da Microsoft Corporation nos Estados Unidos e/ou em outros países.

UNIX é uma marca registrada do The Open Group nos Estados Unidos e em outros países.

Linux é uma marca registrada de Linus Torvalds nos Estados Unidos e/ou em outros países.

Este produto inclui software desenvolvido pelo Projeto Eclipse (<https://www.eclipse.org/>).

Java e todas as marcas registradas e logotipos baseados em Java são marcas ou marcas registradas da Oracle e/ou de suas afiliadas.









Part Number:

(1P) P/N: